UILU-ENG-82-2555

GENERATION EXPANSION PLANNING USING MULTI-ATTRIBUTE UTILITY THEORY AND DYNAMIC PROGRAMMING

E. J. KRULL

Power Affiliates Program

Department of Electrical Engineering

University of Illinois at Urbana-Champaign

Urbana, Illinois 61801

PAP-TR-82-3

September 1982

ACKNOWLEDGEMENTS

I would like to thank Professors R. D. Shultz and R. A. Smith for their time, many ideas and helpful suggestions that made it possible for me to complete this research. I would also like to express my appreciation to the Power Affiliate Program companies. Their contributions made it financially possible for me to attend graduate school.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 Motivation

As construction costs and interest rates increase, it has become necessary to focus more attention on long-range planning of electric generation facilities. A utility has numerous alternatives for expansion when the type of generator to install, the number of generators necessary and the timing of additions are all considered. The environment, system reliability and the cost of installation are all affected by the expansion plan chosen.

Generation expansion planning is one of the many problems that involve multiple and conflicting objectives. To decide which would be the "best" expansion policy, the evaluation of their various attributes and the trade-offs between them must be considered. The trend in generation planning has been to make certain a minimum reserve requirement is met to insure reliability, and that it be done in the most economical way with some attention also focused on the environmental impact. A formal procedure that takes into account reliability, environmental or social effects and economic costs is needed. The procedure could be used by the system planner to help determine the future generation expansion policy of the utility.

This thesis presents a procedure to calculate the optimal generation expansion plan based on an evaluation of economic cost, environmental impact and system operating factors. Using Multi-Attribute Utility (MAU) theory to calculate a generator configuration's overall utility or value, and dynamic programming to project an optimal expansion pattern, an electric

utility's future generation expansion program is determined. The many generation patterns, each having various attributes, are evaluated through the use of a computer program that can be adapted to many existing power systems and their different expansion possibilities. A sensitivity study is then undertaken to see how the choice of different weighting factors affects the optimal expansion plan.

The optimal plan is projected ahead for five years, although this study can be easily adapted for longer planning periods.

1.2 Literature Summary

The use of dynamic programming in generation expansion planning has been a subject of interest for almost twenty years. K. M. Dale wrote a paper [1] in 1966 that outlined a computer program that selected the optimum type and size of a generating unit to be added, based on an evaluation of capital and production costs. He noted that the then present computer size and speed limitations made it necessary to simplify some computations, and to restrict the possible expansion states considered. Although recent computer advancements have made these limitations less restrictive, they will always exist to some extent. However, the number of possible expansion states can be limited by some real world constraints such as capital investment limitations.

E. N. Oatman and L. J. Hamant in 1972 derived their own method [2] to automatically generate the possible expansion patterns. Utilizing a production cost study to limit the number of states, the core memory and central processor unit time requirements were held to reasonable levels.

In 1977, W. F. Esser et al. published a paper in the related field of decision analysis [3]. This paper reviewed a wide range of electrical engineering applications where decision analysis could be helpful in decision making. Each of the situations involved uncertainty and multiple options.

Applications of MAU theory in engineering evaluations were tied together with decision analysis by J. J. Bolinger Jr. and W. F. Esser et al. in 1978 [4]. Using an example of a street-lighting installation problem, a balance between multiple conflicting objectives was shown to be achievable. This procedure of stating the objectives, listing the attributes, defining the utility functions and evaluating the different possibilities can be easily adapted to the generation expansion planning problem.

Earlier work by R. T. Jenkins and D. S. Joy in 1974 showed that dynamic programming could be a valuable tool in generation expansion planning [5]. Their Wein Automatic System Planning (WASP) package derived an optimal expansion plan based on an evaluation of operating and fixed costs. However, no attempt was made to include environmental effects or to similarly include the "value" of system reliability.

The above summaries show that dynamic programming and MAU theory can be useful in situations where multiple options and conflicting objectives are present. This thesis is an attempt to merge these two ideas and use them in a computer program that calculates an "optimal" generation expansion plan.

2. GENERATION CONFIGURATION VALUE OR UTILITY

2.1 Introduction to Multi-Attribute Utility Theory

Multi-Attribute Utility theory is used to evaluate a decision maker's preferences and value tradeoffs in a situation that involves multiple conflicting objectives. Details of the theory and the axioms it is based upon can be found in reference [6]. An example of an electrical engineering application can be found in reference [4].

The main objective of this study is to determine the generation expansion plan which is in the best interests of the electric utility and its customers. The subobjectives and the attributes chosen as a measure for the subobjectives are listed in Table 2.1.

TABLE 2.1. SUBOBJECTIVES AND ATTRIBUTES

Subobjective	Attribute
Reliability of the system	x_1: Loss of Load Probability (LOLP)
Environmental or social benefit	x_2: Impact of fuel on environment
Economic impact on company	x_3: Cost of expansion plan

The form of the multi-attribute utility function must now be determined. Following the same procedure as given in reference [4], the form of the utility function can be determined by testing the attributes for preferential independence, utility independence and additive independence.

Two attributes are defined to be preferentially independent of all other attributes, if the preference order for combinations of those two attributes does not depend upon fixed levels of the other attributes. For example, in this study it was determined by a panel that the preference

order and tradeoffs between x_1 and x_2 were the same for any value when x_3 was held constant. That is, the cost was held at a constant level and the preference order and tradeoffs between the reliability and environmental impact were determined. The cost was then held at a different constant level, and the same preference order and tradeoffs between the reliability and environmental impact were discovered. Next, the preferential independence test was applied to the other attribute pair combinations, and the conclusion was reached that each pair of attributes is preferentially independent of the remaining attribute.

The attributes were then tested for utility independence. An attribute is defined to be utility independent of the others if conditional preferences for that attribute do not depend upon the particular levels at which the other attributes are held constant. For example, the panel's preference structure for various values of the LOLP was determined to be the same no matter what levels the environmental impact and system cost were held at. All three attributes were determined to be utility independent of the others.

The attributes were then tested to see if they displayed the additive independence property. The attributes are defined to be additive independent if preferences over lotteries on these attributes only depend upon their marginal probability distributions. The panel decided that the three attributes used in this study do exhibit the property of additive independence. For example, it was felt that one's preferences for various levels of system reliability are not dependent upon the level of environmental impact. A more reliable system is always preferred to a less reliable system at any level of environmental impact. Similar arguments for other attribute pairings were found.

Knowing that the attributes displayed the properties of preferential, utility and additive independences and using a theorem from reference [6], the form of the multi-attribute utility function is found to be additive. The function is listed below in Equation (2.1.1).

$$U(x_1,x_2,x_3) = \lambda_1 u_1(x_1) + \lambda_2 u_2(x_2) + \lambda_3 u_3(x_3) \qquad (2.1.1)$$

U is the multi-attribute utility function and u_1,u_2,u_3 are the single attribute component utility functions. All of the utility functions are scaled from zero to one, and the λ_i's are scaling constants with $0 < \lambda_i < 1$ and $\sum_{i=1} \lambda_i = 1$. The multi-attribute utility function can be completely defined knowing the scaling constants and the single attribute component utility functions.

The scaling constants can be determined from user preferences with the procedure shown in references [4] and [6]. For this study, the values of the scaling constants were allowed to vary to see how they affected the final results.

The next sections of this chapter develop the single attribute component utility functions. Once these functions are known and the scaling constants are chosen, each possible generation configuration can be evaluated using the multi-attribute utility function. Using the dynamic programming procedure discussed in the next chapter, an optimal generation expansion program can be determined.

2.2 Probabilistic Reliability Evaluation

LOLP was chosen as a good measure of system reliability. To calculate the LOLP, the load profile curve (Figure 2.1) must first be changed to a

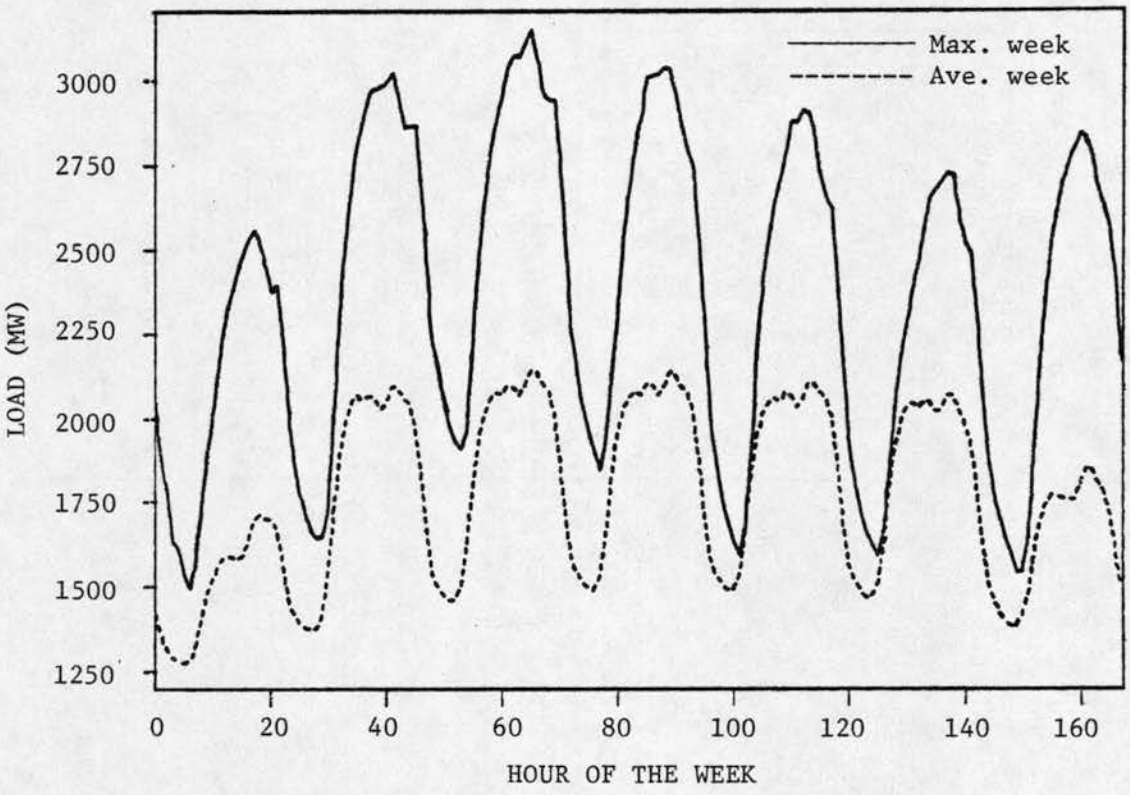

Figure 2.1. Load profile curves for maximum and average load weeks.

load duration curve (Figure 2.2a). This change is done by plotting the percent of time the load is greater than each distinct MW value versus the value of the load in MW. The axes are reversed and time is per-unitized to arrive at the more workable load probability distribution curve, F(L). The curve is then approximated piecewise linear with points chosen where the load probability is 1.0, 0.75, 0.50, 0.25, 0.125, and 0.00 (Figure 2.2b).

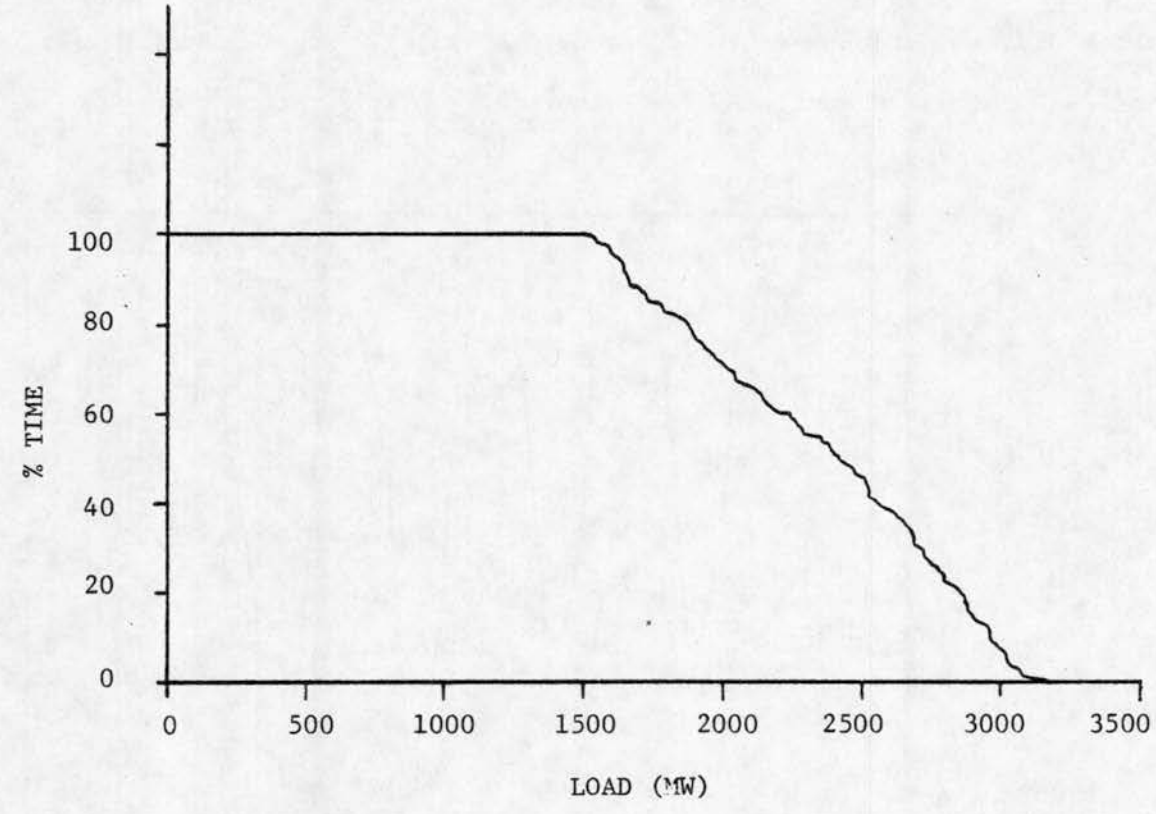

Figure 2.2a. Load duration curve.

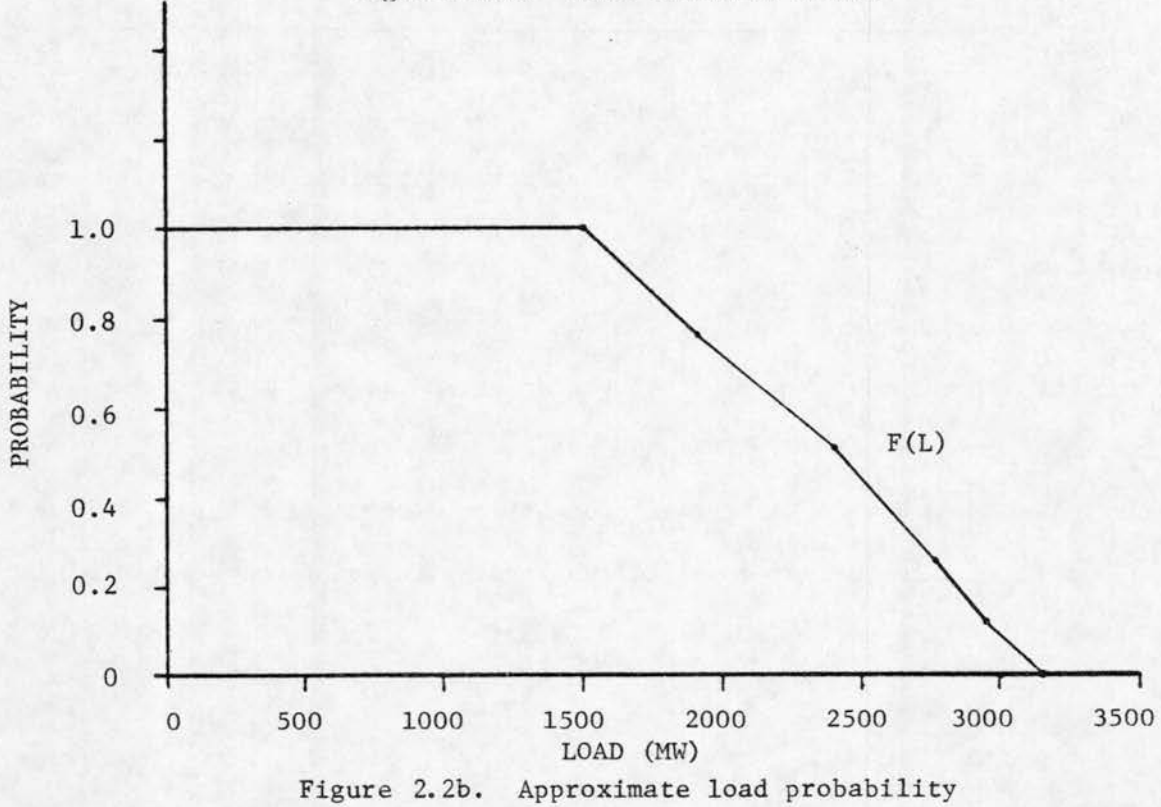

Figure 2.2b. Approximate load probability distribution curve.

By superimposing the generating units in dispatch order on the load distribution curve, (Figure 2.3) and integrating between the proper limits, the energy provided by each unit is determined.

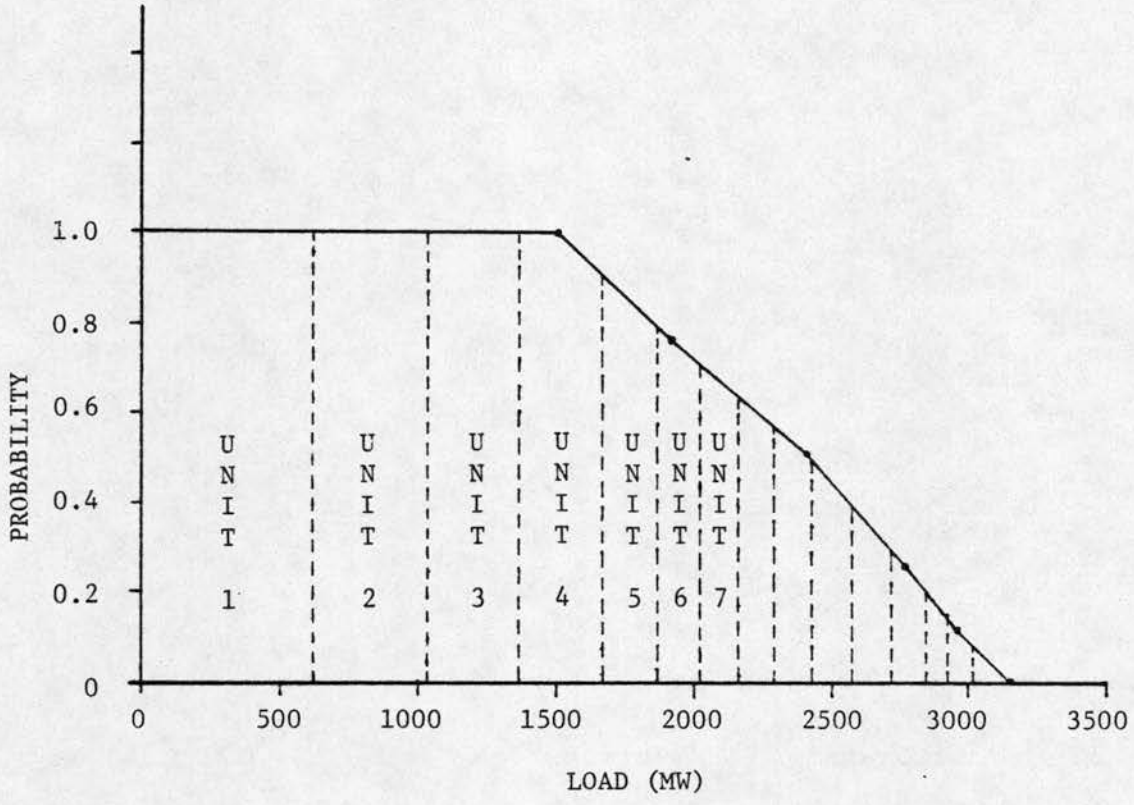

Figure 2.3. Generating units superimposed on the load probability distribution curve.

However, this calculation is only valid for 100% reliable generators. A new load distribution curve must be constructed to account for the forced outages of each generating unit. The system model is shown in Figure 2.4. This model was used by Sullivan in reference [7].

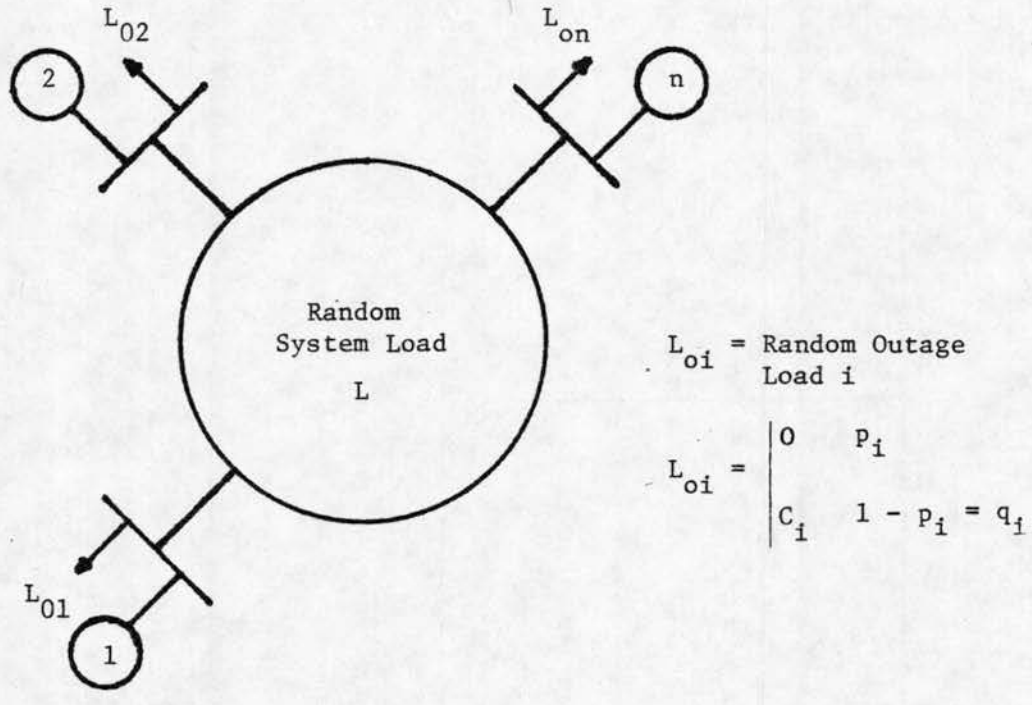

Figure 2.4. System model.

The generating units have been replaced by fictitious 100% reliable generators with random loads equal to the capacity of the generating unit attached to the same bus. The probability density function of each random load is equal to the forced outage probability density function for each respective generator. Therefore, the effective load is given by

$$L_e = L + \sum_{i=1} L_{oi} \tag{2.2.1}$$

where L_{oi} is the random outage load of generating unit i. If C_i represents the generating capacity of unit i, when $L_{oi} = C_i$, the net demand injected

into the system from unit i is zero, just as it should be when there is a forced outage of unit i. The forced outage probability density function is defined as a two-state stochastic process where $f_{oi}(L_{oi} = 0) = p_i$ and $f_{oi}(L_{oi} = C_i) = 1 - p_i = q_i$.

Since L_e is the sum of the independent random variables L and L_{oi} , i = 1 ,..., n, whose distributions are known, $F(L_e)$ can be obtained from the discrete convolution equation

$$F^i(L_e) = \sum_{i=1}^{n} F^{i-1}(L_e - L_{oi})\, f_{oi}(L_{oi}) \qquad (2.2.2)$$

where $F^i(L_e)$ is the effective load probability distribution with the outage capacity of the first i units convolved into the model . Equation 2.2.2 simplifies further to

$$F^i(L_e) = F^{i-1}(L_e)\, p_i + F^i(L_e - C_i)\, q_i \qquad i = 1,..., n \qquad (2.2.3)$$

The effect of the random forced outages on F(L) is to increase the probability that a load will be greater than a given value. The installed capacity is

$$I.C. = \sum_{i=1}^{n} C_i \qquad (2.2.4)$$

the sum of the capacity of each generating unit. After applying Equation 2.2.3 recursively for each generating unit in the dispatch order, the effective load curve is developed (Figure 2.5).

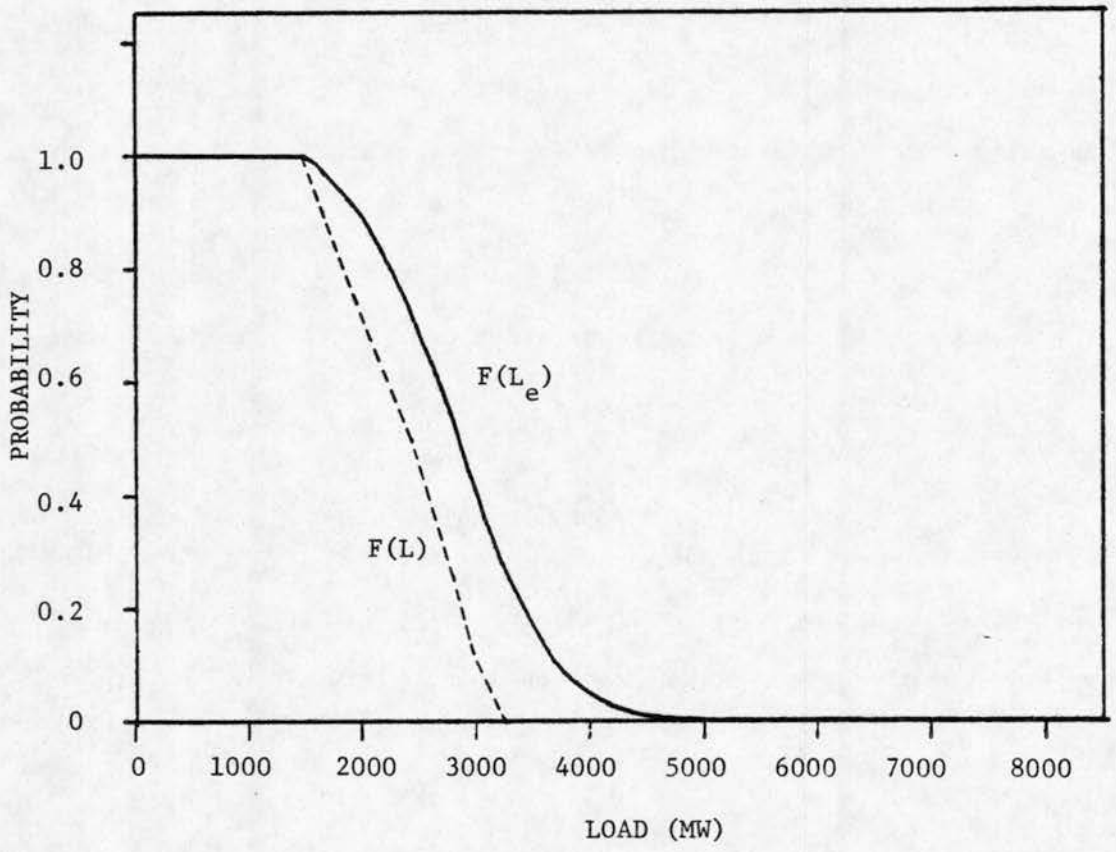

Figure 2.5. Load distribution curve and effective load distribution curve.

The LOLP can be determined by finding the probability of the load when it is greater than the installed generation capacity.

For this study, the generators were allowed to have up to seven different capacity states with each state having a position in the dispatch order. Each capacity state has a forced outage rate that is equal to the forced outage rate of that generator. When a capacity state is convolved into the model, any previously dispatched capacity states for that same generator must be convolved out using

$$F^{i-1}(L_e) = \frac{F^{i}(L_e) - F^{i-1}(L_e - C_i)\, q_i}{p_i} \qquad i = 1, \ldots, n \tag{2.2.5}$$

The larger of those capacity states is then convolved into the model using Equation (2.2.3). This procedure insures that each capacity state of a generator is not interpreted as being a smaller separate generator, with its own forced outage rate.

Before a LOLP for a generation configuration can be calculated, the dispatch order must be known. The generator capacity states are dispatched in the order of least expensive to most expensive according to incremental fuel cost, along with the additional constraint that no generator capacity state can be dispatched until all smaller capacity states of that same generator are dispatched first. For example: if the capacity states of generator 1 are given to be a.) 175 MW, b.) 300 MW, and c.) 525 MW; capacity state (b) cannot be dispatched until (a) has been dispatched. Also capacity state (c) cannot be dispatched unless both (a) and (b) have been dispatched. The incremental fuel cost was calculated using the method shown in Appendix A.

Once all of the generation configuration's capacity states were ordered in this way, a LOLP given in hours/week was calculated for the average load and maximum load week. The average load week and maximum load week probability distribution curves were derived from the hourly load peak data for the entire base year. For the subsequent years of the study, the load data are escalated at a user prescribed load growth rate.

The component utility function for the system reliability was chosen to be a decaying exponential (Equation 2.2.6). For a LOLP = 0. hours/week (the maximum), the benefit was chosen to be the maximum value 1.0.

$$\text{Reliability Benefit} = e^{-c(\text{LOLP})} \qquad (2.2.6)$$

When the LOLP = 168 (hours/week), the worst case, with the proper choice of c, the benefit will be very small. Quantity c was chosen such that a LOLP = 6 (hours/week) yields a reliability benefit of 0.20. (Figure 2.6)

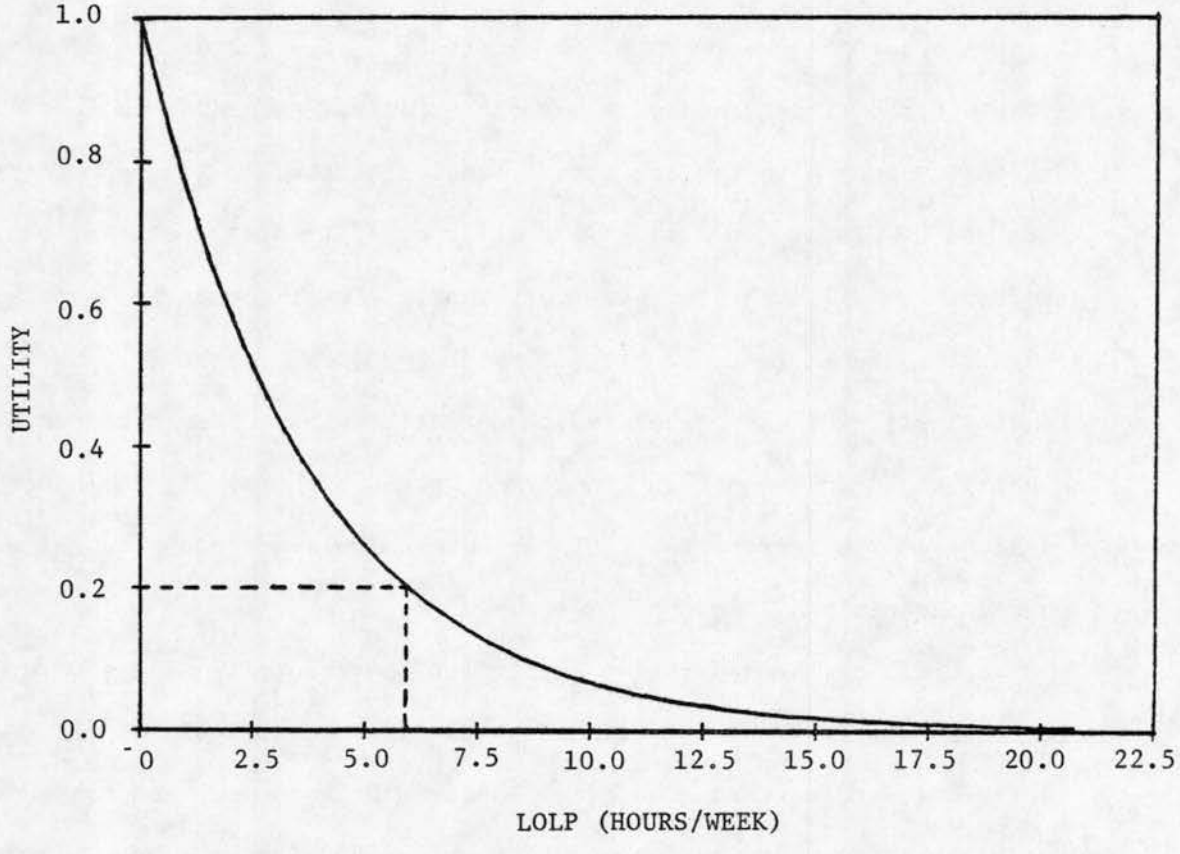

Figure 2.6. Reliability benefit utility function.

Different values of c can be chosen by the user. The total reliability benefit of a generation configuration was defined to be a weighted sum of the benefits determined from Equation (2.2.6), using the LOLP's from the maximum and average load weeks.

$$u_i(x_1) = k_1 e^{-(LOLP_m)c} + k_2 e^{-(LOLP_a)c} \qquad (2.2.7)$$

$u_1(x)$ = reliability component utility function

k_1, k_2 = weighting factors $\quad 0 \leq k_{1,2} \leq 1$, $k_1 + k_2 = 1$

$LOLP_m$ = LOLP for the maximum load week

$LOLP_a$ = LOLP for the average load week

The weighting factors k_1 and k_2 can be chosen by the user. For this study, $k_1 = 0.80$ and $k_2 = 0.20$.

2.3 Environmental Impact Evaluation

The measurement of environmental impact was chosen to be the impact caused by the fuels used in the new generators added to the base system. The fuels used in the possible expansion candidates were ranked by a panel, from 0.0 to 1.0, with 0.0 being the cleanest (smallest impact on the environment). The possible fuels and their environmental impact numbers are listed in Table 2.2.

TABLE 2.2 ENVIRONMENTAL IMPACT NUMBERS

Fuel	Impact
High Sulfur Coal	1.0
Low Sulfur Coal	0.8
Oil	0.3
Natural Gas	0.1

The utility function was chosen to reflect the amount of each fuel used. The capacity of the generator was assumed to be proportional to the

amount of fuel used. The component utility function chosen for the environmental benefit of the generation configuration is given by

$$u_2(x_2) = 1 - \frac{\sum_{i=1}^{m} (C_i)(IM_i)}{MW_b} \qquad (2.3.1)$$

where C_i = capacity of unit i in (MW)

IM_i = impact of unit i

m = total number of newly added units

MW_b = system dependent MW base

The value of MW_b should be chosen so that it is equal to the maximum capacity that the user is willing to add to the system over the length of the study. The environmental benefit does not include any effects of the base system generators because this will be a constant throughout the whole study. Table 2.3 shows some sample calculations for the environmental benefit utility function - given that units A,B,C,D have capacities 400,200,200, and 100 MW with impacts 1.0,1.0,0.5, and 0.3,respectively. MW_b is assumed to be 1000 MW.

TABLE 2.3. SAMPLE ENVIRONMENTAL BENEFIT CALCULATIONS

Unit(s) Installed	Total Capacity	Benefit
B	200	0.800
C	200	0.900
2D	200	0.940
A	400	0.600
B+C	400	0.700
2C	400	0.800
4D	400	0.880
2A	800	0.200
B+3C	800	0.500
3B+C	800	0.300
8D	800	0.760
2A+B	1000	0.000
5C	1000	0.500
10D	1000	0.700

Looking at the table and choosing a value for the installed capacity, the environmental benefit increases as more units with less hazardous fuels are used. The unit with the least hazardous fuel is unit D. For each value of installed capacity, the best that can be done is to install only generating unit D. Notice how the corresponding benefit values decrease with an increase in capacity.

A closer look at Equation (2.3.1) shows that if MW_b were $MW_b = \sum_{i=1}^{m} C_i$, the environmental component utility function would not show that increased capacity would have more environmental impact. In fact, all expansion configurations that are multiples of each other (e.g., B,2B, and 3B) would then have the same environmental benefit. This effect is not desirable. The choice of MW_b being equal to a constant value (the maximum additional capacity being considered) in effect "per-unitizes" the environmental benefit.

2.4 Economic Cost Evaluation

A maximum capital constraint is specified by the user. This maximum capital available is escalated at a rate specified by the user over the subsequent years of the study. The economic benefit component utility function is given by

$$u_3(x_3) = 1 - \frac{\text{capital spent}}{\text{capital maximum}} \qquad (2.4.1)$$

The capital spent is the total amount of money it will cost to build the possible expansion pattern. The cost is escalated, with a user supplied construction escalation rate, from the base year to the present year.

The XDATA input file contains the costs for the expansion candidates and includes all costs connected with construction of the generating unit. Figure 2.7 shows a plot of the utility function. Notice that when the capital spent equals the capital maximum, the benefit equals 0.00, and when the capital spent equals 0.0, the benefit equals 1.0.

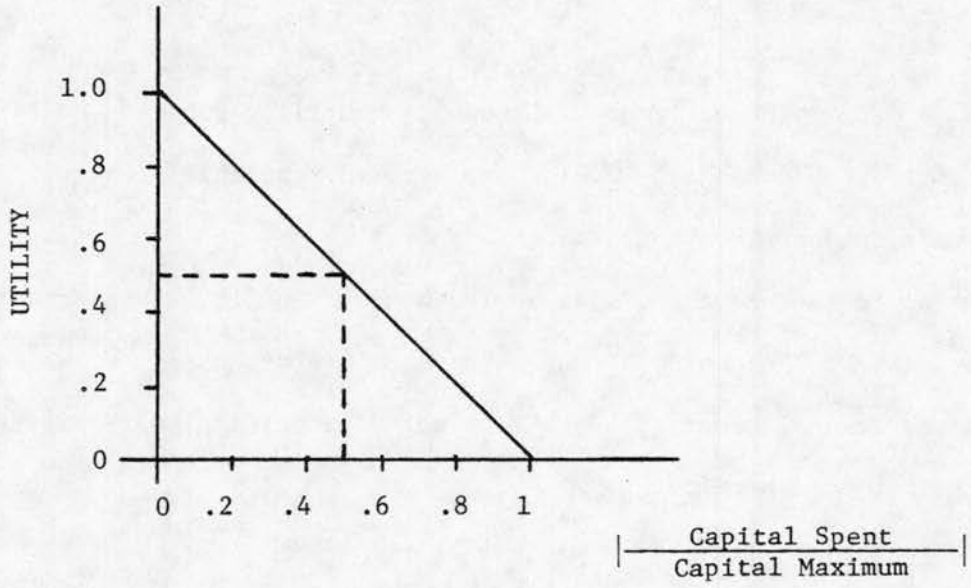

Figure 2.7. Economic benefit utility function.

The component utility functions given by Equations (2.2.7,2.3.1, 2.4.1) were chosen to reflect the feelings of the panel involved. These functions can be changed by other users to reflect their own feelings and preferences. Also, the attributes used to measure the subobjectives and the number of of subobjectives may be changed. References[4] and [6] give the details for developing other utility functions. The usefulness of this kind of study is directly dependent upon the accuracy of the utility functions in modelling the feelings and preferences of the users involved.

3. CALCULATION OF OPTIMAL GENERATION EXPANSION PLAN

3.1 Introduction to Dynamic Programming

This introduction is intended to give the reader enough information to be able to understand the dynamic program used in this research. The interested reader can find more details about dynamic programming in reference [8].

As Bellman stated in reference [8], most problems that can be solved using dynamic programming have the following features:

1. A physical system characterized at any stage by a small set of parameters called the state variables.
2. At each stage, a number of options can be chosen.
3. The effect of a decision is a transformation of the state variables.
4. The past history of the system is of no importance in determining future actions.
5. The purpose of the process is to maximize some function of the state variables.

The generation expansion planning problem has these characteristics. The stages of the problem are defined as the years of the expansion study. The base year of the study is stage 0, with the subsequent years corresponding to stages 1,2,3,...,M. For this research, a five-year planning study is undertaken. The base year is 1980 (stage 0) with the years 1981,82,83,84, and 85 corresponding to stages 1,2,3,4, and 5, respectively.

The state of the generation system at any stage is defined in terms of the state variables $y_1, y_2, y_3, \ldots, y_n$ where y_i = the number of units of expansion candidate (i) installed, i = 1,2,...,n. The four expansion candidates chosen for this study and their capacities are listed in Table 3.1.

TABLE 3.1

EXPANSION CANDIDATES

Type of Unit	Capacity	State Variables
Base Generator	475 MW	y_1
Cyclic Generator	102 MW	y_2
Peaking Generator	48 MW	y_3
Purchase Power	100 MW	y_4

The four state variables are limited in value to the maximum number of each unit the planner is willing to install, as he specifies in the XDATA input file. (See Appendix B.) If $y_{i_{max}} \leq 9$ for all i, then an n digit number can represent the state of the system. For example, in this study $y_{i_{max}} < 9$ for i = 1,2,3,4 is true. The state of the system can be written as the four digit number $y_1y_2y_3y_4$. The benefit of using this representation for the generation system state will be seen later. Using this method, 0000 represents the base system (nothing added) and 1012 represents the base system plus 1 base generator, 0 cyclic generators, 1 peaking generator and 2 purchase power contracts.

At each stage (year) a choice is made regarding the number of each expansion candidate to install. These chosen numbers can also be written as a four digit number $z_1z_2z_3z_4$. ($z_1z_2z_3z_4$ represents an expansion option.) The number of possible expansion options will be very large unless some limiting constraints are added. The first constraint has already been

mentioned above. The number of each expansion candidate (i) is limited to $y_{i_{max}}$. Another constraint is the amount of capital that can be spent on an expansion pattern. The expansion option is first tested to see if it violates a user prescribed maximum capital constraint. If it does, it is not considered as a possible expansion option.

The effect of a decision at each stage is a simple transformation of the state variables. The transformation is as follows: Given the state of the system is $y_1y_2y_3y_4$ and the chosen expansion option is $z_1z_2z_3z_4$ the new state is given by

$$y_1'y_2'y_3'y_4' = y_1y_2y_3y_4 + z_1z_2z_3z_4 \qquad (3.1.1)$$

where $y_i' = y_i + z_i \qquad i = 1,2,3,4$

For example, if the state of the system is 0000 and expansion option 1012 is chosen, the new state is calculated from 3.1.1 as 1012. Or if the state of the system is 1011 and the expansion option chosen is 1210, the new state is 2221. Notice this calculation will not work if the maximum allowed number of any expansion unit is greater than nine; however, this condition is not the case for most planning studies.

The fourth feature listed by Bellman is only partially met by this study. That is, the past history of the system does indirectly affect future actions. Past decisions have helped determine the present state and not all states are accessible from all other states because this study does not allow for the retirement of power plants. However, Bellman notes that the features he listed should not be thought of as strict guidelines.

The optimal generation expansion plan should maximize the

multi-attribute utility function. The component utility functions are a function of the system configuration, and the system configuration is defined by the state variables. Therefore, the goal of this study is to maximize the utility or value as a function of state variables. Using this formulation, the dynamic programming procedure can solve for an optimal expansion plan.

Dynamic programming is based on the Principle of Optimality:

> An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision [8].

The principle of optimality can be illustrated by an example given by Larson in reference [9]. In Figure 3.1, assume the solid line between points R and S is the optimal path between these points, i.e., the lowest cost path. Let T represent any intermediate point on the optimal path. The

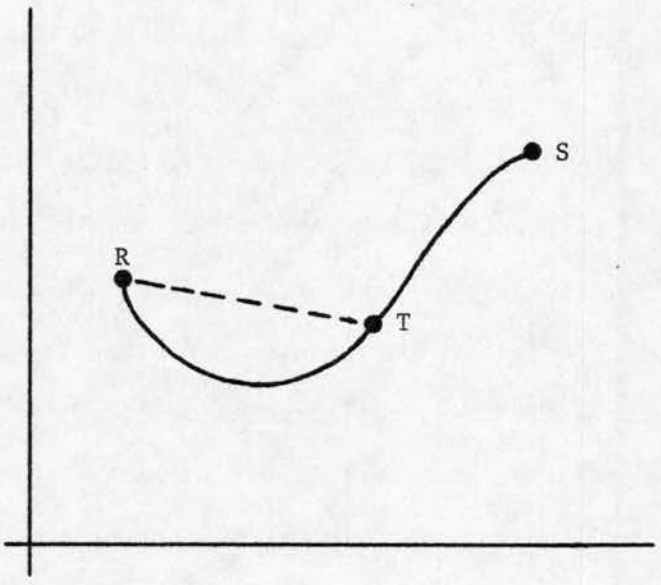

Figure 3.1. Example illustrating the principle of optimality.

portion of the solid line path from R to T must also be the optimal path from R to T. If this condition were not true, then some other path between R and T would exist that would have some lower cost. Let the dashed line between R and T represent such a path. The existence of this path violates the original assumption that the solid line path from R to S is the lowest cost path. A lower cost path would be the dashed line path from R to T continuing with the solid line path from T to S. Therefore, to find the optimal path from R to S, only the optimal path from R to each intermediate point needs to be found.

As given in reference [8], to find the optimal path from R to S, the principle of optimality can be applied by finding the optimal trajectory to every state in each stage. The state in the previous stage that lies on the optimal trajectory to the state under consideration is saved. To find the optimal path to any state in stage j, only the optimal paths to all states in stage j-1 with their associated values for the utility function and the cost (benefit) of going from each of these states to the state under consideration in stage j need to be known. By recursive application of this principle, the optimal path to any state in any stage can be determined. The global optimum over the study period is associated with that state in the final stage which maximizes the utility function, and the optimal path is determined by tracing backward to the initial state.

This procedure is best illustrated by an example. Section 3.3 illustrates the process of tracing through the optimal path. Other examples can be found in references [8] and [9].

3.2 Description of Computer Programs

The "optimal" generation expansion plan is calculated by the two computer programs developed for this research, PREXPAN and XPAN. Program PREXPAN sorts through all of the information contained in the input files COSDAT and DATIN to create smaller data files and also calculates the load probability distribution curves. Program XPAN uses these smaller data files and load probability distribution curves to perform the bulk of the generation expansion planning study.

Both programs are divided into subroutines with each subroutine having a specific task. The calling structures of the programs are shown in Figures 3.2 and 3.3. The subroutine names and functions are listed in Table 3.2.

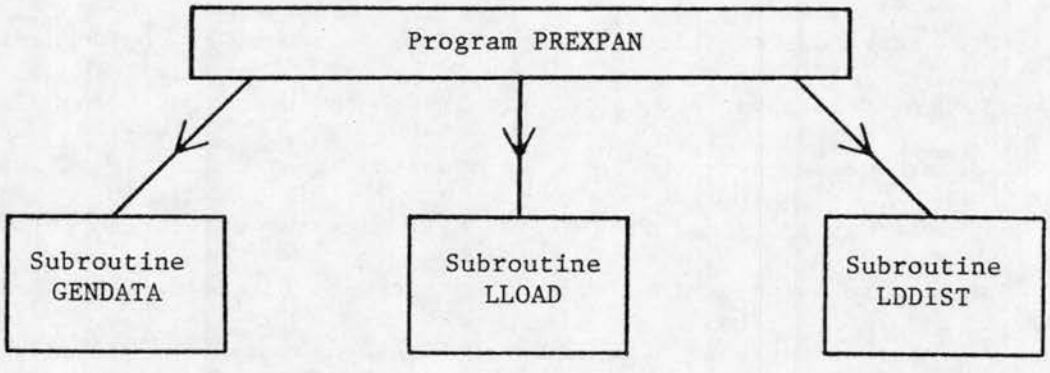

Figure 3.2 Calling structure for program PREXPAN.

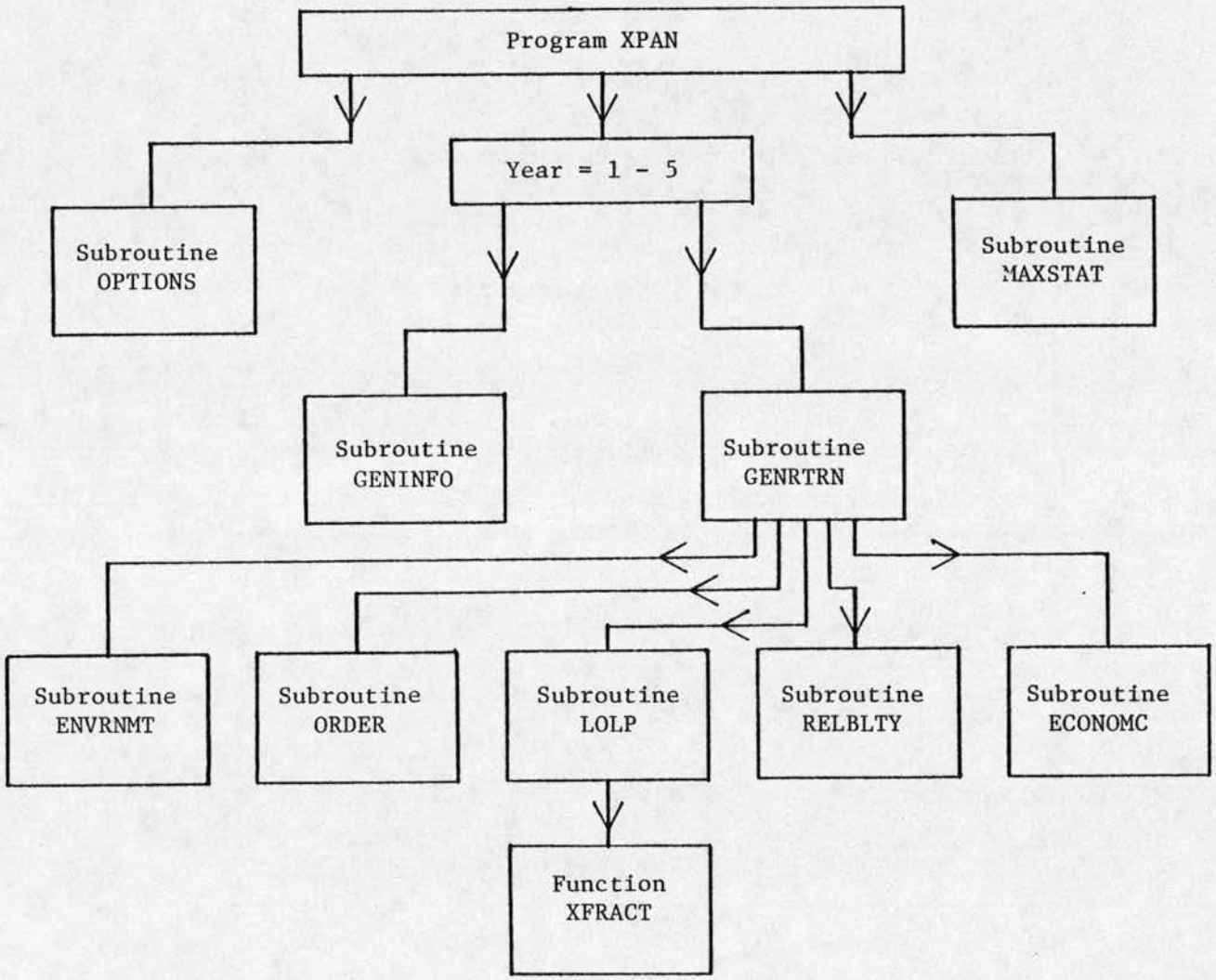

Figure 3.3 Calling structure for program XPAN.

TABLE 3.2

SUBROUTINES AND THEIR FUNCTIONS

Subroutine Name	Called From	Subroutine Function
GENDATA	PREXPAN	Sorts the COSDAT information into smaller files for use by other subroutines
LLOAD	PREXPAN	Finds the maximum load week hourly peaks and calculates the average load week hourly peaks for the base year
LDDIST	PREXPAN	Builds the load probability distribution curves for the maximum and average load weeks in the base year
OPTIONS	XPAN	Finds all possible expansion options that meet the maximum number of units and capital spending constraints
GENINFO	XPAN	Uses the generator data sorted by GENDATA to calculate the incremental fuel cost for each unit capacity state in the current year
GENRTRN	XPAN	Calculates the component utility funtions' values or returns for each expansion option in the current year
ENVRNMT	GENRTRN	Calculates the environmental component utility function's value for each option in the current year
ORDER	GENRTRN	Orders the unit capacity states for each option in the current year by incremental fuel cost
LOLP	GENRTRN	Calculates the maximum and average load weeks' LOLP using the dispatch order given by subroutine ORDER
XFRACT	LOLP	Calculates the value q for every step of the discrete convolution given by Equation (2.2.3)

TABLE 3.2 (Continued)

Subroutine Name	Called From	Subroutine Function
RELBLTY	GENRTRN	Calculates the reliability component utility function's value for every option in the current year
ECONOMC	GENRTRN	Calculates the economic component utility function's value for every option in the current year
MAXSTAT	XPAN	Calculates the multi-attribute utility for all expansion options in every year and saves the maximum return for each expansion option and the previous states that lead to these maximum return options

To do an expansion planning study, the user must first find the component utility functions that reflect his preferences and value tradeoffs. Then all of the necessary information for the computer programs must be gathered and entered into the input files. Program PREXPAN is compiled and executed, thereby creating the local files needed for the execution of program XPAN. Program XPAN is then compiled and executed. The user is prompted by the program to either enter weighting coefficients for the multi-attribute utility function or to stop execution. If the user chooses to enter weighting coefficients, then the program will calculate the multi-attribute utility for all possible expansion options and print out all necessary information for tracing the "optimal" path in the DATOUT file. The user is then prompted again to either enter another set of weighting coefficients or to stop execution. After the program finishes execution, the user prints the output file DATOUT and traces the "optimal" path(s) using the procedure given in the next section.

The restrictions for this computer study were chosen to keep core memory and central processor unit time requirements small but still large enough to allow a detailed study. The restrictions are:

1. A maximum of four generating units can be chosen as expansion candidates.
2. A maximum of thirty expansion states or options can be considered in any year.
3. The maximum number of unit capacity states for any generation configuration is 100.
4. The expansion study is limited to five years.

These restrictions can be charged to fit the individual needs of the user.

3.3 Tracing the "Optimal" Expansion Plan

Using the example output given in Table 3.3 the "optimal" path will be traced. The expansion state numbers are listed across the top and the beginning state numbers are listed on the left side. The multi-attribute utility or return in going from any beginning state to any expansion state is found by reading the proper element from the return matrix. At the bottom of the return matrix for each year, the expansion state numbers are repeated followed by the maximum return possible for each expansion state, and the beginning state that gives this maximum return. The format codes for the return values and state numbers are F4.1 and I4, respectively. Before year one, the weighting coefficients are printed. After year five, the best expansion state and the overall utility for the "optimal" path are printed.

TABLE 3.3

SAMPLE OUTPUT USED TO TRACE THE "OPTIMAL" PATH

ENVIRONMENTAL WT.= .2000 RELIABILITY WT.= .6000 ECONOMIC WT.= .2000

********** THE RETURN VALUES FOR YEAR 1 **********

EXPANSION STATES-->	0	1	2	10	11	12	20	21	22	100	101	102	110	111	112	120	121	122	200	201	202	210	211	212	220	221	222100010011002
0	.93	.93	.92	.91	.90	.89	.89	.87	.86	.88	.87	.86	.85	.84	.83	.82	.81	.80	.82	.81	.79	.79	.78	.76	.76	.74	.74	.67	.65	.64

XPN STATE	0	1	2	10	11	12	20	21	22	100	101	102	110	111	112	120	121	122	200	201	202	210	211	212	220	221	222100010011002
MAX RTRN	.93	.93	.92	.91	.90	.89	.89	.87	.86	.88	.87	.86	.85	.84	.83	.82	.81	.80	.82	.81	.79	.79	.78	.76	.76	.74	.74	.67	.65	.64
BEG STATE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

********** THE RETURN VALUES FOR YEAR 2 **********

EXPANSION STATES-->	0	1	2	10	11	12	20	21	22	100	101	102	110	111	112	120	121	122	200	201	202	210	211	212	220	221	222100010011002
BEGINNING STATES																											
0	1.83	1.82	1.82	1.80	1.80	1.81	1.78	1.78	1.78	1.78	1.78	1.78	1.75	1.76	1.74	1.73	1.73	1.71	1.73	1.73	1.71	1.71	1.69	1.68	1.68	1.66	1.65	1.58	1.57	1.56
1	0.00	1.83	1.83	0.00	1.81	1.81	0.00	1.78	1.78	0.00	1.78	1.78	0.00	1.76	1.75	0.00	1.73	1.72	0.00	1.73	1.72	0.00	1.70	1.69	0.00	1.67	1.65	0.00	1.57	1.56
2	0.00	0.00	1.82	0.00	0.00	1.80	0.00	0.00	1.77	0.00	0.00	1.77	0.00	0.00	1.74	0.00	0.00	1.71	0.00	0.00	1.70	0.00	0.00	1.67	0.00	0.00	1.64	0.00	0.00	1.55
10	0.00	0.00	0.00	1.81	1.81	1.81	1.79	1.79	1.78	0.00	0.00	0.00	1.76	1.76	1.75	1.74	1.74	1.72	0.00	0.00	0.00	1.72	1.70	1.69	1.69	1.67	1.65	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	1.80	1.81	0.00	1.78	1.78	0.00	0.00	0.00	0.00	1.76	1.74	0.00	1.73	1.71	0.00	0.00	0.00	0.00	1.70	1.68	0.00	1.66	1.65	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	1.79	0.00	0.00	1.77	0.00	0.00	0.00	0.00	0.00	1.73	0.00	0.00	1.70	0.00	0.00	0.00	0.00	0.00	1.67	0.00	0.00	1.64	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	1.79	1.79	1.79	0.00	0.00	0.00	0.00	0.00	0.00	1.74	1.74	1.72	0.00	0.00	0.00	0.00	0.00	0.00	1.69	1.67	1.66	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.77	1.77	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.72	1.71	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.66	1.64	0.00	0.00	0.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.76	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.63	0.00	0.00	0.00
100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.78	1.78	1.78	1.76	1.76	1.75	1.74	1.73	1.72	1.74	1.73	1.72	1.71	1.70	1.68	1.68	1.67	1.65	0.00	0.00	0.00
101	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.77	1.77	0.00	1.75	1.74	0.00	1.72	1.70	0.00	1.72	1.70	0.00	1.69	1.67	0.00	1.65	1.64	0.00	0.00	0.00	0.00
102	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.76	0.00	0.00	1.73	0.00	0.00	1.69	0.00	0.00	1.69	0.00	0.00	1.66	0.00	0.00	1.63	0.00	0.00	0.00
110	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.75	1.76	1.74	1.73	1.73	1.71	0.00	0.00	0.00	1.71	1.69	1.68	1.68	1.66	1.65	0.00	0.00	0.00
111	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.74	1.73	0.00	1.72	1.70	0.00	0.00	0.00	0.00	1.68	1.67	0.00	1.65	1.64	0.00	0.00	0.00
112	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.72	0.00	0.00	1.69	0.00	0.00	0.00	0.00	0.00	1.66	0.00	0.00	1.63	0.00	0.00	0.00
120	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.73	1.72	1.71	0.00	0.00	0.00	0.00	0.00	0.00	1.67	1.66	1.64	0.00	0.00	0.00
121	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.71	1.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.65	1.63	0.00	0.00	0.00
122	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.62	0.00	0.00	0.00

TABLE 3.3 (Continued)

```
 200   0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.001.721.721.701.701.681.671.671.651.640.000.000.00
 201   0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.001.711.690.001.671.660.001.641.630.000.000.00
 202   0.000.000.000.000.000.000.000.000.000.000.600.000.000.000.000.000.000.000.000.001.680.000.001.650.000.001.620.000.000.00
 210   0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.001.691.681.661.661.651.630.000.000.00
 211   0.000.000.000.000.000.000.000.000.000.060.000.000.000.000.000.000.600.000.000.000.000.001.671.650.001.641.620.000.000.00
 212   0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.001.650.000.001.620.000.000.00
 220   0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.001.661.641.630.000.000.00
 221   0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.001.631.620.000.000.00
 222   0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.001.610.000.000.00
1000   0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.001.491.481.47
1001   0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.001.471.46
1002   0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.001.45
------------------------------------------------------------------------------------------------------------------------------
XPN STATE    0    1    2   10   11   12   20   21   22  100  101  102  110  111  112  120  121  122  200  201  202  210  211  212  220  221  22210001001 1002
MAX RTRN 1.831.831.831.811.811.811.791.791.791.781.781.781.761.761.751.741.741.721.741.731.721.721.701.691.691.671.661.581.571.56
BEG STATE    0    1    1   10   10   10   20   20   20  100    1    1   10   10   10   20   20   20  100    1    1   10   10   10   20   20   20    0    1    1

      ***************************************** THE RETURN VALUES FOR YEAR  3 *****************************************
EXPANSION
STATES-->    0    1    2   10   11   12   20   21   22  100  101  102  110  111  112  120  121  122  200  201  202  210  211  212  .20  221  22210001001 1002

BEGINNING
  STATES

   0   2.702.692.682.682.672.662.652.642.642.652.642.632.622.612.612.602.592.592.602.592.592.572.572.572.542.542.542.472.452.44
   1   0.002.692.690.002.672.660.002.642.640.002.642.640.002.622.620.002.592.590.002.592.590.002.572.570.002.552.540.002.452.44
   2   0.000.002.690.000.002.670.000.002.640.000.002.640.000.002.620.000.002.600.000.002.600.000.002.580.000.002.550.000.002.44
  10   0.000.000.002.692.672.672.662.652.640.000.000.002.632.622.622.612.602.600.000.000.002.582.572.582.552.552.540.000.000.00
  11   0.000.000.000.002.672.670.002.652.640.000.000.000.002.622.620.002.602.600.000.000.000.002.572.580.002.552.540.000.000.00
  12   0.000.000.000.000.002.680.000.002.650.000.000.000.000.002.630.000.002.610.000.000.000.000.002.580.000.002.550.000.000.00
  20   0.000.000.000.000.000.002.662.652.650.000.000.000.000.000.002.612.502.600.000.000.000.000.000.002.562.562.550.000.000.00
  21   0.000.000.000.000.000.000.002.662.650.000.000.000.000.000.000.002.612.610.000.000.000.000.000.000.002.562.550.000.000.00
  22   0.000.000.000.000.000.000.000.002.650.000.000.000.000.000.000.000.002.610.000.000.000.000.000.000.000.002.550.000.000.00
 100   0.000.000.000.000.000.000.000.000.002.662.652.642.632.622.622.602.602.602.602.602.602.582.572.572.552.552.540.000.000.00
 101   0.000.000.000.000.000.000.000.000.000.002.652.640.002.622.620.002.602.600.002.602.600.002.582.580.002.552.550.000.000.00
```

TABLE 3.3 (Continued)

```
 102  0.000.000.000.000.000.000.000.000.000.000.002.650.000.002.620.000.002.600.000.002.600.000.002.580.000.002.550.000.000.00
 110  0.000.000.000.000.000.000.000.000.000.006.000.002.632.632.622.612.602.600.000.000.002.582.582.582.552.552.550.000.000.00
 111  0.000.000.000.000.000.000.000.000.000.000.000.002.632.630.002.612.610.000.000.000.002.582.580.002.562.550.000.000.00
 112  0.000.000.000.000.000.000.000.000.000.000.000.000.000.002.620.000.002.600.000.000.000.000.002.570.000.002.540.000.000.00
 120  0.060.000.000.000.000.000.000.000.000.000.000.000.000.000.002.612.612.610.000.000.000.000.000.002.562.562.550.000.000.00
 121  0.000.000.000.000.000.000.000.000.000.000.000.000.060.060.000.002.612.610.000.000.000.000.000.000.002.562.550.000.000.00
 122  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.002.590.000.000.000.000.000.000.000.002.540.000.000.00
 200  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.002.612.602.602.582.582.582.552.552.550.000.000.00
 201  0.000.000.000.030.000.000.000.000.000.000.000.000.000.000.000.000.000.000.002.602.600.002.572.580.002.552.550.000.000.00
 202  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.002.590.000.002.560.000.002.530.000.000.00
 210  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.002.592.582.582.562.562.550.000.000.00
 211  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.002.572.570.002.552.540.000.000.00
 212  0.000.000.000.000.000.000.000.000.000.000.000.060.000.000.000.000.000.000.000.000.000.000.002.560.000.002.530.000.000.00
 220  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.002.562.562.550.000.000.00
 221  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.002.542.540.000.000.00
 222  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.002.530.000.000.00
1000  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.002.402.382.37
1001  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.002.382.36
1002  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.002.36
---------------------------------------------------------------------------------------------------------------------------
XPN STATE    0    1    2   10   11   12   20   21   22  100  101  102  110  111  112  120  121  122  200  201  202  210  211  212  220  221  222100010011002
MAX RTRN 2.702.692.692.692.672.682.662.662.652.662.652.652.632.632.632.612.612.612.612.602.602.592.582.582.562.562.552.472.452.44
BEG STATE    0    1    2   10   11   12   20   21   22  100  101  102  110  111   12  120  120   22  200  101  102  210  111   12  120  120   22    0    1    2

     ****************************************** THE RETURN VALUES FOR YEAR   4 ******************************************
EXPANSION
STATES-->    0    1    2   10   11   12   20   21   22  100  101  102  110  111  112  120  121  122  200  201  202  210  211  212  220  221  222100010011002

BEGINNING
  STATES

   0  3.503.523.543.493.513.513.483.503.493.483.503.483.473.473.463.463.443.433.463.443.433.433.413.403.402.393.383.303.303.30
   1  0.003.513.530.003.513.500.003.493.480.003.493.480.003.463.450.003.443.420.003.433.420.003.413.400.003.383.370.003.303.30
   2  0.000.003.530.000.003.500.000.003.480.000.003.470.000.003.450.000.003.420.000.003.420.000.003.400.000.003.370.000.003.30
  10  0.000.000.003.503.523.523.493.513.490.000.000.003.483.483.463.463.453.440.000.000.003.433.423.413.413.393.390.000.000.00
  11  0.000.000.000.003.513.510.003.503.480.000.000.000.003.473.460.003.443.430.000.000.000.003.413.400.003.393.380.000.000.00
```

TABLE 3.3 (Continued)

```
  12  0.000.000.000.000.003.510.000.003.490.000.000.000.000.003.460.000.003.430.000.000.000.000.003.410.000.003.380.000.000.00
  20  0.000.000.000.000.000.003.493.513.500.000.000.000.000.000.003.473.453.440.000.000.000.000.000.003.413.403.390.000.000.00
  21  0.000.000.000.000.000.000.003.513.490.000.000.000.000.000.000.003.453.440.000.000.000.000.000.000.003.393.390.000.000.00
  22  0.000.000.000.000.000.000.000.003.490.000.000.000.000.000.000.000.003.440.000.000.000.000.000.000.000.003.390.000.000.00
 100  0.000.000.000.000.000.000.000.000.003.483.503.483.433.473.463.463.443.433.463.443.433.433.423.413.403.393.380.000.000.00
 101  0.000.000.000.000.000.000.000.000.000.003.503.480.003.473.460.003.443.430.003.443.430.003.413.400.003.393.380.000.000.00
 102  0.000.000.000.000.000.000.000.000.000.000.003.480.000.003.450.000.003.430.000.003.430.000.003.400.000.003.380.000.000.00
 110  0.000.000.000.000.000.000.000.000.000.000.000.003.483.483.463.463.453.440.000.000.003.433.423.413.403.393.390.000.000.00
 111  0.000.000.000.000.000.000.000.000.000.000.000.000.003.483.460.003.453.440.000.000.000.003.423.410.003.393.390.000.000.00
 112  0.000.000.000.000.000.000.000.000.000.000.000.000.000.003.460.000.003.440.000.000.000.000.003.410.000.003.390.000.000.00
 120  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.003.473.453.440.000.000.000.000.000.003.413.403.390.000.000.00
 121  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.003.453.440.000.000.000.000.000.000.003.403.390.000.000.00
 122  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.003.440.000.000.000.000.000.000.000.003.390.000.000.00
 200  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.003.463.443.433.433.423.413.403.393.380.000.000.00
 201  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.003.443.430.003.413.400.003.393.380.000.000.00
 202  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.003.430.000.003.410.000.003.380.000.000.00
 210  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.003.443.423.413.413.403.390.000.000.00
 211  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.003.423.410.003.393.390.000.000.00
 212  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.003.420.000.003.390.000.000.00
 220  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.003.413.403.390.000.000.00
 221  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.003.403.390.000.000.00
 222  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.003.390.000.000.00
1000  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.003.253.253.25
1001  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.003.243.24
1002  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.003.23
----------------------------------------------------------------------------------------------------------------------------
XPN STATE    0   1   2  10  11  12  20  21  22 100 101 102 110 111 112 120 121 122 200 201 202 210 211 212 220 221 222100010011002
MAX RTRN 3.503.523.543.503.523.523.493.513.503.483.503.483.483.493.463.473.453.443.463.443.433.443.423.423.413.403.393.303.303.30
BEG STATE    0   0   0  10  10  10  20  20  20 100 100 100  10  10  10 120 120 122 200 200 202 210 210 212 120 120 212   0   0   0
```

TABLE 3.3 (Continued)

THE RETURN VALUES FOR YEAR 5

BEGINNING STATES \ EXPANSION STATES-->	0	1	2	10	11	12	20	21	22	100	101	102	110	111	112	120	121	122	200	201	202	210	211	212	220	221	222100010011002
0	4.254	.244	.254	.234	.234	.244	.214	.214	.234	.214	.214	.234	.194	.204	.234	.184	.194	.214	.184	.194	.204	.164	.194	.184	.154	.164	.154	.074	.064	.06
1	0.004	.264	.270	.004	.254	.260	.004	.244	.250	.004	.244	.250	.004	.224	.250	.004	.214	.230	.004	.224	.230	.004	.214	.200	.004	.184	.170	.004	.084	.08
2	0.000	.004	.300	.000	.004	.280	.000	.004	.280	.000	.004	.280	.000	.004	.270	.000	.004	.250	.000	.004	.250	.000	.004	.220	.000	.004	.190	.000	.004	.10
10	0.000	.000	.004	.254	.254	.264	.234	.244	.260	.000	.000	.004	.214	.234	.254	.204	.224	.230	.000	.000	.004	.194	.214	.204	.184	.194	.170	.000	.000	.00
11	0.000	.000	.000	.004	.284	.290	.004	.264	.280	.000	.000	.000	.004	.254	.280	.004	.244	.260	.000	.000	.000	.004	.244	.230	.004	.214	.200	.000	.000	.00
12	0.000	.000	.000	.000	.004	.290	.000	.004	.280	.000	.000	.000	.000	.004	.280	.000	.004	.260	.000	.000	.000	.000	.004	.230	.000	.004	.200	.000	.000	.00
20	0.000	.000	.000	.000	.000	.004	.254	.264	.280	.000	.000	.000	.000	.000	.004	.224	.244	.250	.000	.000	.000	.000	.000	.004	.204	.214	.190	.000	.000	.00
21	0.000	.000	.000	.000	.000	.000	.004	.284	.300	.000	.000	.000	.000	.000	.000	.004	.264	.270	.000	.000	.000	.000	.000	.000	.004	.234	.220	.000	.000	.00
22	0.000	.000	.000	.000	.000	.000	.000	.004	.290	.000	.000	.000	.000	.000	.000	.000	.004	.260	.000	.000	.000	.000	.000	.000	.000	.004	.200	.000	.000	.00
100	0.000	.000	.000	.000	.000	.000	.000	.000	.004	.244	.254	.274	.224	.244	.264	.214	.234	.244	.214	.234	.244	.204	.224	.214	.194	.204	.180	.000	.000	.00
101	0.000	.000	.000	.000	.000	.000	.000	.000	.000	.004	.274	.290	.004	.264	.280	.004	.254	.260	.004	.254	.260	.004	.244	.230	.004	.224	.200	.000	.000	.00
102	0.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.004	.280	.000	.004	.270	.000	.004	.250	.000	.004	.250	.000	.004	.220	.000	.004	.190	.000	.000	.00
110	0.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.004	.254	.264	.294	.244	.254	.270	.000	.000	.004	.224	.254	.244	.214	.224	.210	.000	.000	.00
111	0.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.004	.264	.290	.004	.254	.260	.000	.000	.000	.004	.254	.230	.004	.224	.210	.000	.000	.00
112	0.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.004	.280	.000	.004	.260	.000	.000	.000	.000	.004	.230	.000	.004	.200	.000	.000	.00
120	0.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.004	.254	.274	.280	.000	.000	.000	.000	.000	.004	.234	.244	.220	.000	.000	.00
121	0.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.004	.264	.270	.000	.000	.000	.000	.000	.000	.000	.004	.234	.210	.000	.000	.00
122	0.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.004	.260	.000	.000	.000	.000	.000	.000	.000	.004	.200	.000	.000	.00
200	0.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.004	.244	.254	.264	.224	.254	.244	.214	.224	.210	.000	.000	.00
201	0.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.004	.244	.250	.004	.244	.220	.004	.214	.200	.000	.000	.000	.00
202	0.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.004	.250	.000	.004	.220	.000	.004	.190	.000	.000	.000	.00
210	0.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.004	.234	.254	.244	.224	.234	.210	.000	.000	.00
211	0.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.004	.244	.230	.004	.224	.200	.000	.000	.00
212	0.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.004	.230	.000	.004	.200	.000	.000	.00
220	0.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.004	.224	.210	.000	.000	.00
221	0.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.004	.210	.000	.000	.00
222	0.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.004	.200	.000	.00

TABLE 3.3 (Continued)

```
1000  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.00 .000.000.004.064.054.04

1001  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.004.054.05

1002  0.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.004.05

-------------------- ------------------------------------------------------------------------------------------------------------------------------------------------------
XPN STATE   0   1   2  10  11  12  20  21[ 22] 100 101 102 110 111 112 120 121 122 200 201 202 210 211 212 220 221 222100010011002
MAX RTRN 4.254.264.304.254.284.294.254.284.304.244.274.294.254.264.294.254.274.284.244.254.264.234.254.244.234.244.224.074.084.10
BEG STATE   0   1   2  10  11  12  20  21[ 21] 100 101 101 110 110 110 120 120 120 200 200 200 210 210 210 120 120 120   0   1   2
********** THE BEST EXPANSION OPTION IN YEAR 5 IS STATE:  [22] THE OVERALL UTILITY FOR THE OPTIMAL PATH IS: 4.2898650
```

To trace the "optimal" path, first look at the bottom of the return matrix for the last year of the study. The computer program has found that expansion state 0022 has the maximum return value 4.2998650. The beginning state of year five that gives this return value is state 0021. The beginning state of year five is the expansion state of the previous year, year four. Now looking at the output of year four it is found that beginning state 0020 gives the maximum return for expansion state 0021. Looking at year three, beginning state 0020 gives the maximum return for expansion state 0020. The same is true for year two. Now looking at year one, it is found that the only beginning state possible is state 0000 (the base generation configuration). The "optimal" path is listed in Table 3.4.

TABLE 3.4

"OPTIMAL" PATH

Year of study	0	1	2	3	4	5
State of system at end of year	0000	0020	0020	0020	0021	0022

Different weighting coefficients in Equation (2.1.1) may cause the "optimal" path to change, but it is always found using this procedure.

4. CONCLUSIONS AND RECOMMENDATIONS

Results of the computer simulation show different weighting coefficients for the multi-attribute utility function do cause the "optimal" expansion plan to change. As more importance is placed on system reliability, the expansion plan tends to dictate the addition of more units. If instead the economic weighting coefficient is increased, the expansion plan tends toward the addition of smaller, cheaper generating units (if any). If the environmental weighting coefficient is increased, the generators with the cleanest fuels tend to be added. Changing the base generator configuration, load growth rate, construction cost escalation rate or capital escalation rate also affects the expansion plan.

A study such as this could be very helpful to the system planner if the utility functions and weighting constants accurately reflect the feelings of those involved. Although the computer program developed for this research does work as is, it should be improved if it is to be used in the future.

This study did not allow for the retirement of any generating units. If the study is expanded past ten or fifteen years, some of the units which make up the base system might be considered for retirement. An additional constraint should be included that cancels purchase power contracts when large units are added.

The economic component utility function should be altered so that the spending of capital is not so severely penalized. Some results have shown that the economic component utility function restricts expansion even on systems with smaller reserve margins and/or higher

load growth rates. The actual shape may be closer to that of the load probability distribution curve.

The environmental component utility function should be altered if the retirement of units is allowed. Some older units burning dirtier fuels or having inferior pollution control devices may be retired thereby helping the environment.

The reliability component utility function may be altered to include the expected demand not served by the generation configuration as another measure of system reliability. The LOLP calculation might also be changed so that scheduled maintenance outages are not treated as random forced outages.

Sparse matrix techniques should also be included to keep core requirements minimal. Not all expansion states are accessible from all beginning states, therefore, many of the elements in the return or benefit matrices are equal to zero. When the retirement of units is not considered, over sixty percent of the elements will be zero.

Multi-attribute utility theory and dynamic programming can be combined and used by the system planner to aid in generation expansion planning. A rational and quantitative decision can be arrived at when these two techniques are effectively combined in a planning study such as the one derived for this research. However, care should be taken when defining the component utility functions and choosing the weighting coefficients so that the user's actual preferences and value tradeoffs are accurately reflected.

APPENDIX A

INCREMENTAL FUEL COST CALCULATION

For each generator, the unit heat rate is either modelled using a quadratic curve or entered directly as incremental heat rates for each capacity state. An input file holds the necessary information. The equation for the quadratic curve is:

$$y = c_1 + c_2 x + c_3 x^2 \tag{A.1}$$

where

y = unit heat rate in (MBtu/hr)

x = capacity in (MW)

c_1 = constant coefficient in (MBtu/hr)

c_2 = linear coefficient in (MBtu/MW·hr)

c_3 = quadratic coefficient in (MBtu/MW^2·hr).

The equation for the unit heat rate, given the incremental heat rates, is

$$y = \frac{z}{1000} x \tag{A.2}$$

where y and x are defined above and z = incremental heat rate in (BTU/kW·hr). Now using the other information that must be supplied by the user, the incremental fuel cost is calculated. If

$Cost_1$ = the fuel cost in (¢/MBtu)

$Cost_2$ = the fixed operating and maintenance costs of a unit in ($1000/week)

$Cost_3$ = the variable operating and maintenance costs of a unit in ($/MW·hr)

then

$$\text{Fuel Cost}_{incr} = \frac{(.01)(Cost_1)y + \frac{1000}{168}(Cost_2)}{x} + Cost_3 \qquad (A.3)$$

The incremental fuel cost for every generation capacity state is calculated this way.

APPENDIX B

INPUT FILE INSTRUCTIONS AND DATA

The data used in this study are typical for power companies in the Midwest. The computer programs developed for this research have three data input files. They are named COSDAT, DATIN and XDATA. The details for the development of these files are given next.

The COSDAT file is divided into seven groups, which are listed in Table B.1. All of the data cards contain the four letter group code in

Table B.1

COSDAT INPUT DATA GROUPS

Group Code	Information
FUEL	Characteristics of fuel type
FCST	Fuel cost in ¢/MBtu
LLIB	Hourly load data
UBAS	Generating unit information
UCAP	Unit capacity states
UFOR	Unit forced outage rates
UHRT	Unit heat rate characteristics

columns 1-4, except for the LLIB cards. (explained later) All cards from the same group must be placed together, with no other cards from a different group in between. Each grouping of cards begins with a "leader" card (even the LLIB group). This card specified the group code in the first four columns, and all other columns are blank. All of the cards from that group then follow the leader card. After the last card in a group, the next group's leader card is placed, followed by its own group cards. The terminating card of the COSDAT deck is a card that contains

"LAST" in columns 1-4. The groups can be in any order, as long as the leader cards separate the groups and the terminating card is placed at the end.

The details of each COSDAT group are given in the following Tables B.2-B.8.

Table B.2

FUEL GROUP DATA CARDS

Column	Format	Variable
1-4	A4	Group code "FUEL"
5	1X	(not used)
6-7	I2	Fuel type number
8-9	2X	(not used)
10-17	A8	Fuel type name
18-19	2X	(not used)
20-23	A4	Units of fuel (e.g., tons, gals.)
24-25	2X	(not used)
26-34	F9.2	Fuel Heat Conversion factor (MBtu/unit of fuel)
35-36	2X	(not used)
37-41	F5.1	Per-cent fuel cost escalation/ year for this fuel

Table B.3

FCST GROUP DATA CARDS

Column	Format	Variable
1-4	A4	Group code "FCST"
5	1X	(not used)
6-7	I2	Year
8-9	2X	(not used)
10-11	I2	Fuel type number. Must match a number from columns 10-11 of FUEL group.
12-16	5X	(not used)
17-23	F7.1	Fuel cost for the year given in columns 6-7 (¢/MBtu)

Table B.4

LLIB GROUP DATA CARDS

Column	Format	Variable
1-2	I2	Month
3-4	I2	Day
5-20	16X	(not used)
21-80	12F5.0	12 hourly load values

The LLIB group has some other restrictions. Two data cards must be included for each day of the base year. The first contains the am loads, (12 midnight to 1 am, 1 am to 2 am ... 11 am - 12 noon) the second card contains the pm loads. (12 noon - 1 pm ... 11 pm - 12 midnight). The days must be placed in the proper order (January 1 - December 31). A leader card must be included with "LLIB" in columns 1-4, and a special terminating card that contains the number 99 in columns 1-2 must be included after the last (December 31) card. Cards with only the number 98 in columns 1-2 can be placed after the last card for each month to assist in data entry or data checking.

Table B.5

UBAS GROUP DATA CARDS

Column	Format	Variable
1-4	A4	Group code "UBAS"
5	1X	(not used)
6-13	A8	Plant name
14-15	I2	Unit number
16-17	2X	(not used)
18-20	I3	Fuel type number (see FUEL record)
21-22	2X	(not used)
23-27	F5.2	Fixed operating and maintenance cost ($1000/week)
28-29	2X	(not used)
30-35	F5.3	Variable operating and maintenance cost ($1/MW·hr)

Table B.6

UCAP GROUP DATA CARDS

Column	Format	Variable
1-4	A4	Group code "UCAP"
5	1X	(not used)
6-13	A8	Plant name
14-15	I2	Unit number
16	1X	(not used)
17	I1	"1" if expansion, "0" if base unit
18-23	F6.0	Unit capacity state 1 (MW)
24-71	6(2X,F6.0)	Unit capacity states 2-7 (MW) (0.0 if not needed)

Table B.7

UFOR GROUP DATA CARDS

Column	Format	Variable
1-4	A4	Group code "UFOR"
5	1X	(not used)
6-13	A8	Plant name
14-15	I2	Unit number
16-17	2X	(not used)
18-22	F5.1	Unit forced outage rate (%)

Table B.8

UHRT GROUP DATA CARDS

Column	Format	Variable
1-4	A4	Group code
5	1X	(not used)
6-13	A8	Plant name
14-15	I2	Unit number
16-17	2X	(not used)
18	A1	Heat rate modelling option: "I" incremental heat rates are entered, "Q" input/output curve is described by a quadratic question
19-20	2X	(not used)
21-26	F6.0 or F6.2	Incremental heat rate at first capacity state for corresponding unit in (Btu/kW•hr) or constant coefficient (MBtu/hr)
27-28	2X	(not used)
29-35	F7.0 or F7.4	Incremental heat rate at second capacity state for corresponding unit in (Btu/kW•hr) or linear coefficient (MBtu/MW•hr)
36-37	2X	(not used)
38-44	F7.0 or F7.5	Incremental heat rate at third capacity state for corresponding unit in (Btu/kW•hr) or quadratic coefficient in ($MBtu/MW^2$•hr)
45-76	4(2X,F6.0)	Incremental heat rates at 4th, 5th, 6th, 7th capacity states for corresponding unit all in (Btu/kW•hr)

The DATIN input file consists of only one data card and it contains the information listed in Table B.9.

Table B.9

DATIN INPUT FILE

Column	Format	Variable
1-2	I2	Base year of study (e.g.,80)
3	1X	(not used)
4-9	F6.4	Load growth rate (e.g., 0.0370)
10	1X	(not used)
11-17	I7	Maximum capital available ($1000's) in base year
18	1X	(not used)
19	I1	Day number for Jan. 1 in base year (e.g., Sun=1, Mon=2)
20	1X	(not used)
21	I1	Day number for Dec. 31 in base year
22	1X	(not used)
23-28	F6.4	Construction cost escalation rate (e.g., 0.0800)
29	1X	(not used)
30-35	F6.1	Maximum increment for equivalent load convolution (MW)
36	1X	(not used)
37-42	F6.4	Capital escalation rate (e.g., 0.0300)
43	1X	(not used)
44-48	F5.0	Maximum MW considered for expansion
49	1X	(not used)
50-56	F8.5	LOLP scaling constant c (see Eq. 2.2.7)
57	1X	(not used)
58-62	F5.3	LOLP weighting constant k_1 (see Eq. 2.2.7)
63	1X	(not used)
64-68	F5.3	LOLP weighting constant k_2 (see Eq. 2.2.7)

The XDATA input file contains information about the possible expansion units that is not contained in the input file COSDAT. Each unit has its own data card and they should be ordered from largest capacity to smallest. Table B.10 contains the details for the XDATA cards.

Table B.10

XDATA DATA CARDS

Column	Format	Variable
1-7	I7	Total unit construction cost in ($1000's)
8	1X	(not used)
9-10	I2	Maximum number of this unit willing to install
11	1X	(not used)
12-13	I2	Environmental impact for this unit
14	1X	(not used)
15-18	I4	MW capacity of this unit
19	1X	(not used)
20-27	A8	Plant name for this unit
28	1X	(not used)
28-29	I2	Plant number for first new unit installed

The actual data input to the computer program are listed next.

INPUT FILE DATIN

```
80 0.0370  370000 3 4 0.0800 1000.0 0.0600 0700. 45.06426 0.800 0.200
```

INPUT FILE XDATA

```
323500  1 10 0475 PLANT A   3
 90500  2  8  102 PLANT D   2
 46900  2  7   48 PLANT E   5
  5200  2  5  100 PURCHASE  1
```

INPUT FILE COSDAT

```
FUEL
FUEL   2  NEWFOSSL  TONS      21.60   12.5
FUEL   3  PLANT A   TONS      21.70    9.0
FUEL   4  PLANT C   TONS      21.70    9.0
FUEL   5  HSC       TONS      21.60    9.0
FUEL   6  LSC       TONS      22.00    9.0
FUEL   7  LSC       TONS      25.50    9.0
FUEL   8  OIL  NO6  GALS        .14    9.0
FUEL   9  OIL  NO6  GALS        .15    9.0
FUEL  10  GAS       MCF       1.03    9.0
FUEL  11  GAS       MCF       1.03    9.0
FUEL  12  OIL       GALS        .14    9.0
FUEL  18  STARTOIL  GALS        .14    9.0
FCST
FCST  80   2        135.0
FCST  81   2        151.9
FCST  82   2        170.9
FCST  83   2        192.2
FCST  84   2        216.2
FCST  85   2        243.3
FCST  80   3        110.6
FCST  81   3        114.7
FCST  82   3        128.6
FCST  83   3        144.2
FCST  84   3        161.5
FCST  85   3        182.4
FCST  80   4        152.2
FCST  81   4        145.1
FCST  82   4        153.4
FCST  83   4        175.5
FCST  84   4        200.8
FCST  85   4        229.7
FCST  80   5        125.9
FCST  81   5        134.3
FCST  82   5        146.9
FCST  83   5        162.1
FCST  84   5        177.1
FCST  85   5        196.2
FCST  80   6        185.7
FCST  81   6        208.2
FCST  82   6        237.1
FCST  83   6        267.2
FCST  84   6        300.0
FCST  85   6        337.5
FCST  80   7        197.1
FCST  81   7        232.3
FCST  82   7        269.0
FCST  83   7        325.7
FCST  84   7        373.0
FCST  85   7        450.5
FCST  80   8        609.5
FCST  81   8        664.3
FCST  82   8        724.1
FCST  83   8        724.1
FCST  84   8        724.1
FCST  85   8        935.2
FCST  80   9        343.2
FCST  81   9        407.0
FCST  82   9        509.8
FCST  83   9        608.9
FCST  84   9        683.2
```

```
FCST 85   9        798.1
FCST 80  10        280.0
FCST 81  10        325.0
FCST 82  10        373.7
FCST 83  10        429.8
FCST 84  10        494.2
FCST 85  10        538.5
FCST 80  11        210.7
FCST 81  11        230.0
FCST 82  11        269.0
FCST 83  11        303.0
FCST 84  11        347.0
FCST 85  11        378.2
FCST 80  12        549.0
FCST 81  12        699.0
FCST 82  12        838.8
FCST 83  12       1006.5
FCST 84  12       1207.8
FCST 85  12       1449.3
FCST 80  18        549.0
FCST 81  18        699.0
FCST 82  18        838.8
FCST 83  18       1006.5
FCST 84  18       1207.8
FCST 85  18       1449.3
LLIB
98 0                  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 1 1               1358.1328.1286.1273.1263.1266.1307.1310.1358.1429.1495.1508.
 1 1               1537.1539.1506.1512.1582.1687.1650.1621.1589.1548.1453.1373.
 1 2               1359.1365.1369.1343.1366.1439.1604.1793.1879.1956.2001.2006.
 1 2               1999.2002.2015.1996.2066.2151.2122.2047.2010.1937.1807.1634.
 1 3               1553.1511.1506.1496.1517.1569.1747.1912.1997.2032.2059.2063.
 1 3               2021.2023.2009.1964.2038.2172.2134.2068.2033.1958.1839.1676.
 1 4               1581.1545.1541.1520.1557.1598.1754.1935.2023.2051.2074.2061.
 1 4               2019.2044.2009.1987.2033.2140.2102.2042.1980.1921.1839.1693.
 1 5               1574.1513.1485.1462.1471.1486.1527.1575.1671.1723.1763.1760.
 1 5               1748.1720.1702.1680.1740.1880.1852.1795.1747.1706.1635.1531.
 1 6               1460.1416.1387.1372.1367.1339.1434.1463.1513.1574.1583.1600.
 1 6               1648.1635.1617.1633.1717.1790.1764.1729.1686.1636.1550.1518.
 1 7               1501.1512.1520.1531.1558.1622.1832.2024.2081.2156.2193.2116.
 1 7               2097.2064.2067.2019.2113.2264.2250.2179.2118.2052.1943.1787.
 1 8               1684.1644.1642.1638.1641.1729.1897.2074.2116.2154.2181.2162.
 1 8               2087.2066.2025.2005.2096.2268.2227.2194.2131.2071.1946.1813.
 1 9               1681.1692.1687.1655.1691.1748.1937.2095.2143.2165.2166.2149.
 1 9               2070.2059.2027.1987.2064.2217.2212.2168.2141.2064.1940.1792.
 110               1682.1660.1656.1643.1679.1723.1884.2045.2078.2106.2126.2106.
 110               2108.2083.2007.1988.2072.2153.2134.2075.2012.1942.1806.1632.
 111               1500.1504.1451.1417.1411.1477.1621.1825.1903.1974.2042.2036.
 111               2014.2009.1979.1950.1988.2177.2171.2115.2043.1999.1899.1760.
 112               1646.1608.1563.1546.1536.1540.1582.1654.1721.1789.1824.1811.
 112               1763.1724.1697.1694.1772.1933.1902.1847.1785.1736.1658.1547.
 113               1462.1421.1397.1381.1378.1400.1420.1451.1527.1582.1605.1627.
 113               1640.1609.1591.1581.1646.1765.1762.1708.1700.1656.1571.1483.
 114               1451.1442.1449.1449.1468.1526.1730.1883.1954.1967.1924.1934.
 114               1876.1873.1813.1793.1847.2035.2047.1976.1962.1879.1816.1599.
 115               1511.1485.1472.1485.1461.1566.1734.1896.2012.2050.2052.2053.
 115               2001.2028.1981.1985.2023.2117.2085.2023.1974.1887.1777.1611.
 116               1551.1498.1484.1472.1484.1543.1716.1897.1957.1998.2004.1984.
 116               1961.1964.1922.1872.1914.2057.2019.2004.1963.1875.1801.1618.
 117               1529.1533.1506.1503.1507.1587.1728.1889.1966.2016.2021.2009.
 117               1973.1965.1955.1944.1974.2114.2102.2063.2014.1926.1833.1681.
```

```
118
1612.1573.1567.1553.1565.1639.1792.1961.2024.2047.2063.2003.
118
1947.1936.1865.1853.1802.1975.1994.1945.1900.1841.1755.1626.
119
1519.1456.1438.1436.1431.1453.1508.1539.1634.1711.1736.1741.
119
1711.1683.1649.1650.1702.1819.1810.1735.1686.1642.1586.1463.
120
1376.1335.1317.1309.1298.1320.1344.1398.1470.1521.1536.1567.
120
1580.1561.1526.1498.1435.1687.1736.1689.1666.1637.1675.1438.
121
1470.1454.1467.1463.1499.1551.1765.1939.2003.2068.2107.2108.
121
2068.2078.2071.2028.2063.2168.2148.2088.2038.1953.1839.1663.
122
1582.1560.1530.1537.1546.1624.1796.1946.2041.2083.2121.2121.
122
2093.2104.2092.2043.2075.2223.2226.2181.2116.2043.1955.1804.
123
1742.1731.1740.1730.1747.1799.1974.2107.2153.2188.2193.2194.
123
2101.2119.2152.2067.2113.2283.2310.2246.2184.2118.1987.1846.
124
1732.1704.1684.1690.1706.1735.1917.2032.2073.2104.2110.2074.
124
2019.1976.1978.1957.1961.2134.2111.2063.2011.1923.1826.1656.
125
1569.1526.1513.1522.1541.1603.1779.1919.1978.2006.2011.2007.
125
1970.1952.1926.1907.1923.2062.2070.2026.1973.1920.1830.1670.
126
1553.1501.1499.1480.1500.1532.1565.1611.1728.1789.1796.1804.
126
1759.1723.1688.1699.1747.1902.1916.1864.1812.1766.1683.1582.
127
1507.1467.1452.1450.1460.1486.1541.1562.1632.1674.1689.1717.
127
1721.1690.1691.1679.1733.1866.1878.1837.1787.1782.1681.1596.
128
1571.1596.1586.1592.1589.1673.1862.2035.2119.2187.2193.2169.
128
2115.2108.2097.2043.2105.2256.2273.2243.2174.2097.1982.1801.
129
1661.1682.1702.1693.1716.1759.1946.2094.2183.2236.2220.2209.
129
2156.2131.2087.2041.2090.2262.2298.2270.2216.2129.2017.1826.
130
1766.1745.1761.1740.1747.1800.1976.2122.2196.2232.2239.2216.
130
2162.2158.2149.2118.2155.2293.2321.2271.2225.2154.2021.1865.
131
1771.1743.1746.1726.1764.1829.2026.2148.2210.2208.2163.2161.
131
2126.2147.2102.2071.2111.2273.2344.2309.2285.2197.2060.1919.
98 0
   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
2 1
1841.1812.1809.1800.1799.1896.2071.2195.2251.2262.2230.2199.
2 1
2122.2115.2063.2029.2051.2210.2246.2207.2139.2073.1976.1835.
2 2
1720.1676.1638.1636.1625.1646.1694.1730.1849.1897.1906.1873.
2 2
1801.1763.1738.1764.1811.1922.1956.1906.1870.1836.1746.1627.
2 3
1561.1519.1513.1490.1503.1508.1553.1581.1646.1647.1633.1618.
2 3
1603.1553.1538.1528.1584.1742.1827.1816.1808.1779.1708.1631.
2 4
1612.1606.1626.1630.1658.1753.1937.2103.2154.2207.2169.2123.
2 4
2077.2068.2044.2027.2052.2176.2245.2196.2162.2094.1956.1796.
2 5
1696.1666.1670.1660.1661.1753.1929.2096.2219.2268.2265.2269.
2 5
2213.2220.2183.2139.2154.2257.2248.2219.2163.2079.1933.1778.
2 6
1692.1663.1659.1645.1675.1745.1899.2030.2109.2173.2180.2165.
2 6
2149.2154.2127.2084.2121.2224.2275.2225.2168.2088.1944.1794.
2 7
1696.1685.1656.1644.1664.1704.1903.2040.2115.2124.2114.2101.
2 7
2034.2026.1990.1765.1998.2137.2200.2166.2124.2050.1906.1772.
2 8
1676.1645.1643.1634.1642.1743.1890.2022.2101.2148.2133.2125.
2 8
2060.2081.2060.2024.2019.2131.2169.2108.2033.1974.1863.1723.
2 9
1597.1561.1552.1528.1535.1554.1610.1666.1759.1820.1824.1797.
2 9
1754.1719.1685.1666.1708.1846.1914.1858.1810.1760.1701.1586.
210
1529.1506.1474.1457.1455.1473.1515.1532.1606.1629.1629.1650.
210
1624.1587.1560.1556.1593.1733.1809.1795.1765.1738.1652.1562.
211
1538.1531.1537.1545.1546.1619.1798.1960.2060.2112.2123.2090.
211
2036.2043.2012.1966.1986.2137.2241.2211.2147.2088.2012.1808.
212
1722.1707.1697.1710.1710.1784.1921.2025.2117.2175.2169.2113.
212
2053.2075.2001.1960.1985.2114.2197.2204.2146.2110.1996.1836.
213
1751.1732.1748.1726.1749.1816.1975.2122.2165.2183.2179.2171.
213
2130.2127.2090.2035.2061.2159.2223.2182.2110.2032.1900.1727.
214
1633.1587.1589.1581.1588.1658.1838.2018.2060.2076.2079.2053.
214
2001.2012.1970.1924.1961.2076.2131.2082.2037.1991.1858.1685.
215
1654.1577.1573.1557.1554.1636.1782.1982.2037.2086.2126.2117.
215
2102.2123.2106.2056.2083.2157.2209.2148.2057.1988.1889.1741.
216
1642.1609.1589.1586.1573.1623.1670.1724.1794.1866.1867.1877.
216
1830.1814.1795.1772.1792.1926.2033.1977.1936.1879.1795.1687.
```

```
217 1620.1573.1554.1556.1559.1582.1609.1607.1653.1655.1655.1661.
217 1630.1593.1548.1537.1561.1669.1799.1786.1768.1757.1670.1604.
218 1556.1577.1568.1579.1596.1663.1812.1951.2041.2077.2068.2028.
218 1989.1996.1944.1925.1959.2066.2137.2094.2059.1969.1835.1666.
219 1581.1557.1537.1528.1541.1594.1762.1937.2011.2052.2054.2033.
219 1997.2004.1987.1960.2009.2080.2126.2077.2018.1955.1812.1673.
220 1592.1582.1575.1546.1549.1606.1780.1939.2014.2000.2060.2016.
220 1992.1986.1961.1922.1943.2027.2105.2054.2009.1958.1810.1668.
221 1564.1519.1510.1522.1521.1578.1760.1906.2013.2083.2102.2063.
221 2032.2024.2012.1964.1953.2029.2067.2021.1965.1888.1783.1584.
222 1534.1521.1489.1489.1505.1560.1726.1867.1954.2000.2027.2016.
222 1988.1993.1949.1906.1890.1951.2004.1947.1877.1829.1718.1591.
223 1467.1429.1387.1385.1360.1393.1431.1474.1584.1678.1705.1733.
223 1691.1676.1648.1650.1680.1754.1788.1739.1689.1627.1557.1455.
224 1371.1333.1301.1294.1287.1293.1326.1345.1420.1463.1489.1499.
224 1484.1464.1445.1423.1447.1516.1647.1644.1639.1602.1532.1461.
225 1431.1427.1422.1437.1459.1560.1703.1890.2042.2148.2195.2186.
225 2167.2152.2125.2079.2090.2164.2250.2247.2192.2124.2005.1843.
226 1760.1738.1737.1700.1725.1787.1948.2052.2125.2151.2161.2153.
226 2124.2127.2107.2089.2109.2165.2296.2231.2181.2094.1959.1795.
227 1712.1687.1677.1647.1660.1719.1869.1982.2054.2076.2075.2053.
227 2025.2012.1986.1934.1966.2042.2130.2088.2050.2002.1898.1731.
228 1636.1606.1590.1595.1615.1636.1817.1937.2026.2052.2047.2039.
228 1980.1991.1972.1966.2002.2060.2177.2170.2092.2041.1937.1777.
229 1700.1667.1675.1658.1697.1747.1880.2072.2130.2139.2135.2134.
229 2046.2096.2073.2074.2116.2197.2264.2218.2169.2091.1997.1875.
98 0    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 3 1 1719.1675.1655.1643.1652.1684.1700.1764.1868.1856.1940.1917.
 3 1 1866.1810.1775.1765.1788.1871.1993.1959.1887.1849.1784.1692.
 3 2 1614.1571.1552.1558.1559.1582.1564.1601.1644.1655.1631.1635.
 3 2 1622.1568.1538.1526.1550.1648.1796.1807.1786.1776.1704.1618.
 3 3 1613.1596.1607.1611.1631.1706.1857.2014.2125.2161.2158.2118.
 3 3 2056.2030.2003.1959.1980.2064.2175.2127.2098.2028.1879.1783.
 3 4 1623.1622.1557.1561.1559.1589.1753.1920.2012.2060.1999.2031.
 3 4 1996.1989.1974.1954.2003.2060.2091.2040.2010.1934.1819.1635.
 3 5 1564.1553.1517.1545.1548.1655.1831.1986.2109.2161.2170.2155.
 3 5 2118.2121.2100.2046.2056.2112.2241.2233.2176.2140.1985.1814.
 3 6 1698.1699.1663.1684.1723.1782.1921.2080.2158.2198.2184.2162.
 3 6 2090.2084.1974.1932.1957.2005.2117.2128.2104.2033.1932.1740.
 3 7 1648.1619.1614.1594.1624.1640.1789.1920.1975.2031.2049.2046.
 3 7 2020.2031.2004.1954.1926.1961.2022.1965.1914.1850.1733.1574.
 3 8 1451.1412.1394.1392.1393.1428.1458.1555.1657.1757.1762.1786.
 3 8 1752.1706.1685.1657.1677.1721.1809.1769.1731.1673.1585.1495.
 3 9 1421.1399.1368.1368.1357.1387.1377.1418.1459.1472.1469.1465.
 3 9 1458.1456.1394.1387.1409.1462.1588.1622.1603.1561.1494.1399.
310 1374.1386.1393.1389.1427.1483.1619.1785.1892.1957.1952.1949.
310 1905.1902.1879.1871.1887.1946.2070.2088.2069.2017.1888.1738.
311 1676.1636.1650.1654.1671.1732.1841.1960.2066.2096.2060.2050.
311 1963.1979.1911.1844.1886.1990.2072.2084.2044.1987.1861.1710.
312 1624.1624.1614.1620.1621.1665.1816.1970.2104.2124.2168.2172.
312 2131.2161.2163.2131.2158.2172.2240.2191.2152.2059.1916.1759.
313 1666.1642.1623.1613.1638.1666.1775.1951.2046.2109.2141.2125.
313 2091.2089.2081.2074.2068.2122.2192.2173.2122.2035.1931.1778.
314 1684.1656.1655.1620.1665.1680.1817.1936.1996.2020.2015.1978.
314 1918.1921.1893.1823.1820.1849.1962.1974.1932.1876.1782.1616.
315 1483.1440.1434.1435.1415.1423.1440.1406.1576.1615.1618.1580.
315 1514.1436.1453.1421.1454.1481.1602.1609.1561.1518.1441.1343.
316 1284.1221.1191.1204.1217.1252.1210.1274.1368.1422.1458.1475.
316 1498.1460.1461.1448.1482.1547.1613.1605.1563.1486.1428.1326.
317 1317.1317.1280.1308.1327.1398.1591.1814.1943.2045.2092.2093.
317 2071.2050.2020.1952.1925.1949.2046.2102.2016.1941.1778.1668.
```

318 1584.1581.1581.1555.1570.1644.1753.1875.1973.2000.1986.1969.
318 1906.1941.1865.1833.1806.1836.1954.2007.1969.1918.1829.1648.
319 1556.1537.1538.1536.1541.1600.1697.1884.1977.2001.2018.1984.
319 1926.1928.1884.1835.1808.1842.1921.1975.1947.1887.1767.1627.
320 1529.1514.1502.1503.1495.1545.1674.1820.1915.1959.1939.1922.
320 1883.1867.1864.1843.1866.1950.2015.1995.1985.1914.1778.1616.
321 1537.1520.1508.1527.1540.1585.1713.1850.1963.2001.2024.1984.
321 1945.1943.1896.1832.1778.1808.1932.1932.1890.1841.1740.1594.
322 1479.1456.1426.1415.1412.1413.1406.1443.1541.1579.1582.1562.
322 1529.1469.1451.1447.1471.1522.1634.1631.1596.1554.1473.1369.
323 1308.1274.1255.1233.1240.1245.1239.1300.1382.1410.1425.1450.
323 1451.1387.1380.1375.1414.1467.1564.1565.1543.1491.1432.1346.
324 1326.1310.1328.1333.1331.1408.1557.1760.1897.1973.2000.1968.
324 1959.1995.1974.1955.1981.2041.2077.2058.2011.1934.1799.1645.
325 1557.1519.1529.1506.1525.1579.1709.1890.2004.2040.2008.2003.
325 1947.1963.1958.1900.1947.1984.2053.2053.2011.1938.1814.1842.
326 1563.1543.1587.1539.1535.1586.1710.1887.1977.2011.2012.2011.
326 1948.1939.1896.1838.1821.1867.1990.1987.1947.1891.1776.1621.
327 1536.1535.1499.1498.1517.1561.1680.1637.1867.1913.1901.1879.
327 1829.1828.1790.1734.1755.1796.1894.1894.1859.1801.1696.1536.
328 1478.1433.1431.1435.1452.1484.1611.1778.1895.1946.1992.1976.
328 1939.1942.1888.1839.1849.1876.1929.1887.1829.1774.1651.1535.
329 1476.1389.1358.1346.1337.1359.1372.1449.1548.1622.1627.1599.
329 1550.1520.1501.1487.1520.1558.1616.1616.1589.1526.1456.1368.
330 1316.1284.1255.1265.1260.1264.1269.1323.1412.1467.1478.1494.
330 1511.1476.1460.1458.1498.1549.1601.1604.1576.1544.1449.1384.
331 1368.1372.1377.1382.1369.1433.1572.1777.1914.1989.2028.2011.
331 1963.1949.1907.1851.1862.1884.1952.1960.1913.1852.1707.1563.
98 0 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
4 1 1467.1442.1426.1426.1429.1495.1598.1757.1845.1885.1880.1877.
4 1 1807.1805.1798.1734.1747.1765.1833.1865.1841.1777.1657.1539.
4 2 1466.1460.1425.1422.1438.1459.1526.1712.1768.1829.1842.1807.
4 2 1785.1782.1761.1721.1728.1730.1834.1881.1850.1794.1679.1551.
4 3 1463.1423.1426.1403.1427.1469.1550.1696.1836.1896.1900.1890.
4 3 1840.1822.1788.1745.1706.1748.1794.1865.1821.1760.1660.1492.
4 4 1391.1349.1351.1341.1354.1386.1422.1514.1627.1707.1748.1726.
4 4 1666.1624.1579.1544.1526.1553.1611.1669.1637.1588.1494.1392.
4 5 1298.1268.1251.1244.1246.1249.1249.1314.1407.1442.1451.1428.
4 5 1381.1343.1329.1293.1316.1352.1472.1535.1498.1443.1379.1268.
4 6 1202.1152.1143.1128.1132.1138.1150.1202.1259.1270.1265.1259.
4 6 1232.1206.1190.1177.1195.1251.1358.1406.1401.1366.1311.1247.
4 7 1228.1216.1229.1222.1229.1268.1358.1550.1705.1766.1815.1790.
4 7 1740.1746.1717.1677.1685.1703.1799.1843.1808.1722.1595.1453.
4 8 1359.1326.1297.1283.1310.1338.1435.1656.1773.1853.1861.1870.
4 8 1826.1837.1827.1801.1814.1871.1936.1925.1880.1818.1717.1537.
4 9 1445.1416.1413.1385.1408.1454.1591.1767.1888.1939.1971.1996.
4 9 1953.1960.1946.1907.1898.1960.1967.1929.1928.1841.1723.1583.
410 1499.1462.1465.1441.1475.1485.1598.1739.1851.1912.1924.1921.
410 1869.1861.1812.1775.1764.1746.1824.1904.1876.1805.1688.1526.
411 1434.1433.1395.1391.1408.1438.1571.1743.1876.1935.1948.1925.
411 1880.1903.1878.1835.1824.1854.1883.1862.1812.1738.1647.1490.
412 1390.1343.1323.1327.1304.1288.1318.1400.1484.1532.1556.1549.
412 1502.1472.1441.1421.1455.1496.1555.1608.1579.1531.1451.1361.
413 1287.1257.1228.1219.1221.1198.1203.1254.1318.1366.1370.1370.
413 1377.1352.1343.1347.1399.1453.1532.1589.1562.1551.1471.1409.
414 1405.1372.1368.1380.1395.1451.1577.1802.1935.1989.2050.2058.
414 2014.2011.1997.1997.1977.2001.2043.2087.2017.1962.1811.1660.
415 1551.1525.1513.1482.1514.1512.1646.1799.1891.1947.1926.1914.
415 1859.1857.1825.1796.1777.1803.1849.1941.1898.1863.1718.1597.
416 1488.1480.1471.1494.1482.1509.1618.1777.1860.1891.1857.1839.
416 1777.1767.1745.1707.1673.1689.1773.1863.1838.1803.1670.1523.

```
417  1438.1411.1399.1407.1414.1417.1577.1751.1860.1895.1880.1879.
417  1798.1816.1799.1774.1758.1802.1856.1915.1868.1785.1648.1493.
418  1411.1391.1380.1375.1392.1394.1519.1690.1792.1803.1830.1819.
418  1763.1792.1772.1685.1649.1640.1651.1775.1729.1658.1550.1390.
419  1291.1238.1237.1208.1189.1156.1210.1292.1384.1443.1457.1446.
419  1419.1390.1380.1368.1379.1393.1423.1514.1477.1404.1330.1223.
420  1157.1103.1074.1062.1067.1021.1057.1118.1211.1259.1281.1305.
420  1298.1309.1278.1270.1290.1297.1350.1472.1466.1424.1315.1212.
421  1179.1166.1173.1170.1185.1179.1348.1537.1670.1758.1805.1820.
421  1793.1820.1858.1783.1763.1766.1762.1875.1841.1754.1593.1420.
422  1300.1277.1250.1239.1261.1244.1409.1606.1736.1820.1853.1874.
422  1898.1880.1896.1876.1878.1857.1846.1952.1909.1826.1649.1477.
423  1357.1312.1284.1281.1264.1248.1413.1600.1715.1783.1803.1814.
423  1799.1810.1784.1739.1717.1711.1755.1825.1816.1717.1588.1424.
424  1309.1290.1274.1261.1269.1266.1382.1620.1737.1778.1816.1803.
424  1766.1770.1746.1673.1669.1677.1729.1832.1811.1743.1613.1455.
425  1368.1330.1326.1313.1330.1388.1471.1688.1709.1755.1765.1741.
425  1701.1693.1689.1628.1611.1615.1666.1758.1710.1651.1538.1404.
426  1301.1248.1234.1208.1216.1185.1242.1312.1390.1443.1447.1438.
426  1390.1382.1362.1343.1378.1401.1436.1503.1470.1419.1323.1244.
427  1160.1149.1138.1122.1111.1100.1100.1132.1207.1261.1286.1309.
427  1320.1295.1291.1265.1285.1319.1342.1359.1435.1410.1359.1266.
428  1233.1233.1227.1235.1264.1314.1452.1666.1800.1842.1891.1827.
428  1818.1823.1797.1748.1762.1755.1771.1793.1815.1763.1627.1488.
429  1345.1328.1312.1309.1336.1387.1501.1694.1844.1901.1905.1884.
429  1867.1890.1827.1800.1778.1782.1767.1763.1829.1790.1664.1492.
430  1397.1359.1338.1343.1343.1395.1516.1699.1793.1868.1863.1852.
430  1813.1811.1786.1738.1742.1744.1736.1711.1831.1780.1651.1450.
98 0    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
5 1  1353.1327.1314.1307.1306.1375.1512.1667.1787.1828.1825.1810.
5 1  1786.1805.1780.1722.1702.1708.1703.1691.1807.1702.1644.1461.
5 2  1347.1312.1291.1285.1294.1347.1442.1639.1718.1829.1820.1841.
5 2  1806.1806.1812.1763.1683.1658.1641.1631.1704.1678.1548.1355.
5 3  1229.1188.1167.1188.1150.1165.1164.1248.1347.1410.1420.1443.
5 3  1425.1396.1381.1388.1393.1413.1420.1390.1474.1443.1355.1236.
5 4  1153.1101.1078.1052.1061.1050.1065.1110.1192.1244.1274.1320.
5 4  1339.1331.1335.1340.1363.1391.1376.1368.1474.1478.1372.1255.
5 5  1208.1187.1169.1162.1177.1227.1330.1538.1655.1774.1836.1845.
5 5  1843.1874.1870.1822.1820.1805.1771.1765.1837.1781.1642.1437.
5 6  1305.1276.1236.1196.1225.1273.1375.1565.1727.1781.1819.1845.
5 6  1819.1841.1828.1800.1779.1763.1727.1754.1811.1781.1616.1457.
5 7  1331.1273.1269.1245.1265.1302.1416.1606.1724.1800.1819.1785.
5 7  1775.1797.1752.1710.1697.1714.1684.1674.1782.1759.1641.1476.
5 8  1368.1327.1322.1315.1328.1373.1469.1671.1787.1840.1869.1861.
5 8  1799.1796.1772.1711.1702.1682.1657.1660.1759.1764.1634.1464.
5 9  1341.1319.1303.1303.1321.1370.1486.1649.1768.1785.1800.1795.
5 9  1752.1771.1729.1688.1641.1638.1611.1601.1688.1655.1538.1307.
510  1247.1192.1175.1161.1159.1173.1176.1238.1362.1426.1464.1427.
510  1436.1401.1366.1363.1364.1385.1372.1370.1447.1434.1355.1241.
511  1160.1119.1082.1057.1063.1040.1024.1097.1183.1243.1262.1284.
511  1275.1278.1254.1258.1265.1271.1282.1321.1400.1407.1350.1239.
512  1200.1190.1190.1173.1179.1237.1349.1550.1712.1824.1891.1855.
512  1822.1845.1833.1777.1752.1755.1748.1781.1814.1757.1629.1451.
513  1317.1292.1270.1245.1264.1291.1385.1513.1765.1824.1891.1881.
513  1864.1876.1849.1805.1743.1730.1692.1684.1766.1726.1594.1424.
514  1315.1282.1262.1265.1256.1318.1424.1596.1700.1762.1779.1782.
514  1724.1762.1756.1717.1670.1644.1612.1608.1698.1701.1577.1406.
515  1303.1257.1251.1250.1268.1280.1408.1582.1693.1753.1741.1749.
515  1741.1770.1752.1707.1705.1687.1646.1654.1734.1702.1607.1426.
516  1297.1243.1240.1221.1233.1261.1281.1570.1694.1756.1769.1791.
516  1774.1772.1758.1701.1685.1694.1671.1666.1672.1626.1512.1310.
```

```
517 1238.1191.1161.1124.1133.1146.1154.1213.1336.1421.1458.1470.
517 1464.1432.1425.1409.1414.1432.1410.1381.1442.1429.1347.1232.
518 1154.1107.1074.1050.1050.1023.1018.1080.1162.1227.1263.1295.
518 1294.1287.1269.1259.1266.1297.1308.1305.1389.1381.1288.1214.
519 1178.1172.1146.1147.1158.1192.1295.1512.1653.1726.1779.1790.
519 1759.1776.1776.1699.1688.1689.1665.1643.1754.1701.1592.1408.
520 1303.1269.1264.1239.1251.1294.1398.1602.1731.1806.1826.1832.
520 1802.1865.1808.1763.1772.1742.1731.1689.1767.1770.1650.1444.
521 1343.1295.1286.1265.1268.1316.1416.1610.1737.1780.1813.1852.
521 1816.1860.1882.1813.1815.1807.1760.1740.1851.1829.1670.1507.
522 1365.1328.1301.1318.1296.1309.1417.1605.1745.1811.1875.1908.
522 1895.1924.1926.1883.1831.1849.1820.1811.1874.1854.1706.1521.
523 1374.1339.1329.1300.1295.1337.1461.1638.1782.1835.1870.1905.
523 1872.1884.1879.1784.1771.1712.1701.1649.1674.1657.1534.1369.
524 1245.1163.1131.1081.1117.1111.1096.1159.1257.1354.1396.1423.
524 1414.1416.1402.1410.1424.1423.1405.1374.1427.1439.1361.1243.
525 1170.1105.1075.1059.1042.1025.1020.1084.1165.1249.1300.1359.
525 1394.1402.1411.1421.1433.1447.1431.1412.1471.1508.1415.1282.
526 1182.1113.1075.1056.1036.1008.1006.1077.1183.1279.1349.1410.
526 1399.1405.1399.1410.1436.1449.1421.1385.1472.1481.1368.1237.
527 1193.1165.1138.1134.1139.1179.1278.1492.1648.1771.1832.1873.
527 1917.1939.1965.1946.1938.1937.1909.1856.1920.1909.1748.1526.
528 1403.1337.1323.1274.1289.1293.1408.1582.1765.1866.1951.2027.
528 2019.2149.2179.2139.2188.2123.2081.2009.1971.2003.1851.1640.
529 1502.1428.1404.1364.1370.1340.1435.1646.1811.1966.2071.2144.
529 2164.2191.2189.2146.2142.2108.2047.1973.1985.1977.1834.1626.
530 1488.1441.1397.1360.1369.1364.1470.1697.1871.1978.2017.2025.
530 2026.2066.2101.2089.2085.2083.2040.1973.1962.1972.1821.1631.
531 1471.1362.1335.1300.1267.1218.1249.1310.1437.1586.1635.1687.
531 1707.1735.1775.1787.1820.1804.1769.1717.1749.1742.1623.1458.
98 0   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
6 1 1327.1273.1218.1193.1174.1189.1126.1212.1303.1379.1463.1470.
6 1 1488.1523.1526.1561.1598.1644.1611.1556.1631.1666.1586.1449.
6 2 1379.1347.1334.1320.1314.1312.1418.1635.1867.2006.2105.2146.
6 2 2115.2082.2080.2049.2063.2042.1999.1987.1994.2003.1845.1670.
6 3 1520.1476.1438.1405.1405.1419.1504.1723.1912.2004.2110.2167.
6 3 2165.2226.2238.2218.2232.2188.2169.2036.2116.2073.1907.1671.
6 4 1529.1430.1385.1364.1363.1346.1419.1622.1764.1828.1881.1880.
6 4 1829.1853.1850.1838.1813.1796.1770.1727.1787.1822.1718.1524.
6 5 1408.1351.1322.1305.1315.1328.1434.1584.1753.1878.1980.2021.
6 5 2096.2218.2282.2301.2312.2344.2320.2250.2248.2270.2129.1832.
6 6 1697.1611.1545.1504.1479.1471.1548.1724.1921.2068.2195.2300.
6 6 2343.2427.2466.2423.2416.2417.2409.2304.2295.2261.2123.1907.
6 7 1724.1625.1549.1497.1471.1420.1441.1500.1727.1875.2013.2047.
6 7 2171.2194.2182.2190.2177.2164.2136.2035.1993.1935.1741.1548.
6 8 1424.1309.1243.1197.1159.1106.1085.1133.1192.1265.1303.1347.
6 8 1338.1329.1305.1298.1317.1324.1337.1302.1337.1408.1339.1233.
6 9 1190.1168.1168.1153.1154.1161.1280.1470.1647.1738.1803.1815.
6 9 1805.1826.1833.1796.1781.1793.1734.1688.1751.1767.1653.1473.
610 1347.1283.1261.1248.1240.1246.1325.1409.1660.1751.1827.1862.
610 1846.1901.1897.1874.1861.1809.1787.1710.1757.1768.1627.1430.
611 1325.1283.1250.1236.1233.1239.1320.1497.1681.1753.1805.1839.
611 1834.1861.1867.1810.1822.1853.1780.1705.1744.1767.1645.1442.
612 1305.1275.1241.1230.1227.1236.1316.1479.1650.1745.1819.1857.
612 1859.1904.1905.1894.1882.1866.1830.1748.1782.1835.1699.1516.
613 1390.1330.1290.1272.1284.1279.1356.1503.1693.1774.1842.1856.
613 1842.1920.1938.1945.1958.1977.1970.1911.1882.1920.1774.1559.
614 1410.1324.1279.1246.1223.1211.1207.1288.1431.1597.1696.1795.
614 1860.1933.2008.2059.2130.2180.2155.2092.2071.2073.1944.1772.
615 1606.1481.1376.1333.1279.1262.1220.1276.1384.1485.1555.1603.
615 1635.1650.1623.1630.1646.1652.1626.1582.1574.1565.1448.1317.
```

```
616          1250.1203.1178.1174.1169.1187.1271.1418.1587.1689.1747.1769.
616          1758.1693.1731.1740.1742.1732.1673.1633.1663.1707.1583.1398.
617          1302.1259.1224.1223.1221.1238.1316.1466.1626.1752.1787.1829.
617          1803.1836.1859.1845.1857.1841.1772.1748.1764.1767.1642.1452.
618          1366.1328.1279.1264.1267.1262.1358.1548.1693.1800.1871.1929.
618          1951.2018.2042.2046.2079.2074.2034.1963.1966.1987.1825.1655.
619          1468.1406.1370.1333.1327.1337.1417.1610.1818.1965.2087.2156.
619          2172.2183.2104.2044.2046.2008.1927.1821.1812.1837.1690.1494.
620          1374.1332.1293.1281.1288.1238.1354.1498.1643.1728.1770.1806.
620          1800.1820.1841.1791.1752.1743.1686.1654.1663.1669.1562.1394.
621          1268.1207.1171.1149.1137.1124.1131.1209.1332.1429.1499.1542.
621          1544.1571.1593.1615.1661.1687.1664.1613.1586.1605.1506.1352.
622          1244.1173.1127.1088.1076.1024.1042.1107.1224.1331.1395.1475.
622          1551.1584.1639.1672.1707.1720.1708.1662.1678.1753.1647.1517.
623          1424.1390.1348.1325.1317.1346.1423.1613.1807.1937.2053.2094.
623          2118.2145.2115.2045.2043.2047.1993.1944.1960.1946.1805.1626.
624          1483.1423.1392.1374.1369.1372.1447.1661.1830.1942.1998.2053.
624          2084.2155.2217.2188.2224.2247.2208.2143.2137.2154.1994.1765.
625          1599.1522.1471.1425.1411.1403.1471.1714.1972.2152.2311.2392.
625          2445.2522.2562.2558.2606.2608.2569.2471.2445.2435.2231.1954.
626          1750.1684.1591.1543.1530.1502.1571.1822.2092.2301.2442.2558.
626          2623.2709.2740.2755.2802.2820.2746.2664.2622.2615.2398.2144.
627          1935.1832.1706.1676.1627.1592.1658.1900.2177.2381.2532.2634.
627          2701.2768.2827.2806.2818.2618.2745.2625.2562.2522.2350.2110.
628          1902.1784.1635.1607.1558.1500.1502.1579.1793.1999.2168.2297.
628          2364.2410.2400.2423.2424.2384.2335.2042.1986.2018.1832.1660.
629          1563.1464.1354.1300.1279.1227.1225.1293.1476.1658.1778.1845.
629          1907.1941.1960.1923.1937.1936.1870.1788.1718.1722.1603.1482.
630          1334.1282.1235.1212.1200.1248.1329.1555.1741.1928.1976.2046.
630          2089.2131.2184.2189.2218.2218.2145.2071.2029.2043.1874.1668.
98 0            0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
 7 1          1465.1382.1405.1375.1365.1353.1447.1643.1858.2002.2143.2295.
 7 1          2388.2547.2640.2700.2782.2804.2796.2752.2741.2744.2570.2307.
 7 2          2111.1979.1908.1812.1750.1737.1815.2009.2193.2298.2403.2488.
 7 2          2564.2649.2662.2597.2482.2309.2230.2147.2120.2061.1927.1733.
 7 3          1601.1508.1456.1395.1404.1394.1493.1668.1802.1902.1936.1956.
 7 3          1990.2078.2143.2168.2211.2238.2214.2160.2117.2104.1975.1807.
 7 4          1605.1511.1449.1396.1379.1353.1313.1357.1507.1672.1849.1949.
 7 4          2032.2076.2123.2174.2222.2243.2229.2157.2112.2088.2038.1920.
 7 5          1839.1734.1641.1535.1383.1284.1251.1395.1587.1789.1953.2082.
 7 5          2188.2242.2299.2339.2392.2395.2369.2292.2225.2208.2057.1861.
 7 6          1706.1560.1442.1381.1343.1284.1270.1343.1487.1637.1755.1870.
 7 6          1971.1958.1952.1938.2037.2079.2065.1985.1985.1978.1856.1669.
 7 7          1554.1475.1446.1401.1362.1382.1502.1766.2073.2301.2522.2680.
 7 7          2768.2884.2932.2949.2960.2949.2854.2848.2830.2802.2633.2332.
 7 8          2133.2014.1913.1858.1815.1795.1882.2137.2427.2601.2770.2862.
 7 8          2909.2970.3025.3048.3026.3002.2965.2901.2855.2835.2591.2332.
 7 9          2126.1978.1899.1822.1775.1752.1827.2035.2258.2407.2484.2571.
 7 9          2638.2769.2827.2888.2932.2979.2905.2838.2832.2818.2611.2379.
710           2184.2048.1954.1856.1855.1843.1904.2037.2203.2358.2479.2608.
710           2705.2800.2870.2895.2955.2937.2881.2783.2751.2749.2543.2283.
711           2076.1945.1883.1769.1747.1713.1785.2026.2303.2533.2683.2772.
711           2873.2917.2928.2953.2918.2884.2828.2737.2683.2662.2484.2212.
712           2062.1995.1943.1863.1792.1768.1748.1887.2086.2283.2454.2589.
712           2655.2721.2765.2805.2836.2829.2757.2663.2603.2575.2414.2190.
713           2045.1881.1784.1636.1608.1537.1493.1564.1718.1871.2002.2148.
713           2278.2353.2414.2477.2525.2558.2526.2450.2379.2396.2240.2024.
714           1881.1783.1724.1659.1648.1650.1737.2007.2286.2527.2676.2817.
714           2884.2963.2977.2983.2997.3022.2958.2865.2867.2866.2678.2419.
715           2238.2137.2047.1975.1930.1909.1958.2192.2464.2628.2791.2880.
715           2952.3035.3075.3071.3110.3150.3069.2969.2940.2941.2754.2515.
```

```
716          2264.2132.2046.1956.1901.1844.1929.2116.2285.2539.2675.2791.
716          2884.3010.3017.3018.3038.3032.2959.2859.2796.2740.2519.2252.
717          2016.1879.1770.1693.1631.1596.1663.1875.2158.2382.2528.2629.
717          2696.2785.2879.2875.2911.2904.2839.2719.2656.2616.2386.2124.
718          1946.1807.1723.1657.1621.1593.1659.1839.2032.2188.2274.2379.
718          2457.2562.2656.2678.2704.2726.2718.2600.2525.2482.2311.2077.
719          1906.1780.1703.1640.1591.1541.1546.1643.1901.2153.2351.2513.
719          2612.2704.2760.2797.2841.2829.2777.2682.2620.2567.2410.2183.
720          1998.1874.1775.1703.1641.1582.1547.1680.1882.2069.2261.2454.
720          2551.2596.2633.2668.2705.2720.2688.2613.2599.2584.2376.2183.
721          2027.1936.1835.1782.1735.1747.1793.1977.2170.2442.2575.2686.
721          2706.2628.2526.2422.2311.2227.2177.2116.2090.2117.1981.1787.
722          1623.1549.1512.1476.1472.1496.1569.1758.1966.2152.2301.2389.
722          2434.2449.2452.2453.2437.2440.2344.2235.2170.2145.1973.1788.
723          1599.1501.1476.1455.1422.1458.1515.1671.1866.2102.2133.2197.
723          2217.2296.2285.2290.2329.2253.2203.2133.2064.2053.1890.1689.
724          1530.1463.1408.1324.1363.1374.1437.1628.1823.1957.2075.2130.
724          2191.2255.2324.2320.2354.2350.2315.2214.2170.2138.1944.1742.
725          1565.1498.1442.1399.1390.1402.1435.1623.1849.2027.2176.2289.
725          2367.2446.2518.2532.2519.2548.2472.2375.2311.2233.2061.1822.
726          1655.1574.1475.1417.1378.1350.1356.1419.1592.1748.1871.1962.
726          2023.2059.2047.1970.1955.1906.1863.1792.1808.1775.1673.1518.
727          1417.1344.1287.1259.1240.1223.1209.1253.1345.1448.1483.1545.
727          1580.1599.1523.1524.1540.1532.1503.1491.1517.1551.1497.1372.
728          1318.1297.1282.1257.1251.1306.1364.1524.1732.1872.1983.2045.
728          2065.2128.2190.2172.2216.2237.2195.2108.2100.2049.1886.1672.
729          1513.1447.1384.1352.1356.1393.1462.1625.1897.2052.2214.2326.
729          2397.2510.2593.2618.2652.2670.2616.2514.2463.2401.2200.1947.
730          1774.1646.1575.1532.1497.1526.1575.1747.1974.2140.2268.2394.
730          2470.2602.2702.2756.2765.2845.2787.2674.2655.2625.2454.2176.
731          2001.1908.1839.1793.1763.1783.1822.1985.2217.2422.2565.2693.
731          2768.2863.2895.2869.2868.2875.2801.2702.2703.2654.2454.2181.
98 0            0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 8 1          1962.1850.1774.1725.1715.1683.1753.1944.2138.2287.2404.2511.
 8 1          2632.2727.2793.2851.2905.2898.2867.2741.2711.2663.2472.2246.
 8 2          2052.1929.1820.1751.1697.1686.1617.1751.1924.2083.2160.2224.
 8 2          2246.2259.2239.2205.2156.2065.1990.1899.1882.1872.1738.1611.
 8 3          1510.1417.1363.1322.1290.1267.1294.1290.1429.1561.1687.1810.
 8 3          1907.1960.2023.2067.2132.2162.2130.2067.2035.2005.1851.1684.
 8 4          1574.1508.1461.1417.1423.1441.1496.1694.1972.2212.2404.2547.
 8 4          2665.2818.2871.2872.2879.2878.2793.2613.2603.2478.2175.1910.
 8 5          1749.1654.1597.1554.1534.1567.1604.1769.1931.2060.2155.2218.
 8 5          2282.2380.2447.2467.2524.2559.2500.2405.2396.2332.2152.1917.
 8 6          1755.1683.1631.1594.1555.1612.1653.1811.2074.2301.2458.2617.
 8 6          2674.2745.2770.2790.2807.2801.2708.2612.2618.2562.2314.2118.
 8 7          1941.1838.1778.1710.1683.1712.1764.1972.2251.2470.2628.2725.
 8 7          2776.2898.2943.2911.2958.2971.2926.2846.2840.2761.2562.2286.
 8 8          2081.1967.1881.1818.1772.1772.1823.2043.2319.2545.2718.2812.
 8 8          2900.2989.3007.3023.3024.3001.2939.2820.2777.2699.2518.2264.
 8 9          2076.1942.1846.1772.1705.1681.1616.1739.1980.2215.2421.2560.
 8 9          2665.2721.2784.2778.2814.2806.2729.2625.2593.2538.2371.2165.
 810          2004.1864.1735.1669.1616.1584.1530.1599.1771.1935.2085.2234.
 810          2358.2439.2499.2533.2564.2577.2518.2401.2356.2186.2007.1864.
 811          1747.1673.1622.1575.1551.1583.1650.1828.2036.2219.2382.2506.
 811          2559.2660.2703.2635.2631.2608.2524.2394.2379.2298.2073.1853.
 812          1699.1598.1533.1498.1468.1475.1550.1743.1992.2165.2272.2266.
 812          2414.2524.2547.2562.2559.2556.2470.2372.2363.2273.2052.1828.
 813          1641.1615.1554.1505.1479.1498.1566.1738.1925.2110.2246.2391.
 813          2489.2552.2555.2516.2522.2505.2431.2342.2350.2268.2105.1889.
 814          1743.1657.1615.1559.1551.1581.1649.1802.1996.2163.2240.2303.
 814          2375.2453.2491.2426.2400.2377.2332.2299.2311.2245.2098.1898.
```

```
 815  1751.1677.1625.1568.1543.1598.1649.1755.1985.2059.2201.2237.
 815  2271.2317.2355.2326.2293.2268.2213.2135.2161.2079.1932.1730.
 816  1556.1502.1431.1389.1369.1367.1385.1410.1526.1650.1715.1793.
 816  1848.1882.1941.1966.1996.2007.1977.1925.1930.1847.1737.1609.
 817  1489.1415.1362.1334.1303.1299.1270.1318.1465.1600.1738.1856.
 817  1942.2066.2135.2194.2250.2282.2275.2205.2214.2161.2023.1822.
 818  1731.1650.1605.1552.1522.1561.1632.1807.1976.2273.2416.2534.
 818  2618.2619.2624.2574.2593.2611.2571.2452.2479.2403.2199.1966.
 819  1792.1698.1645.1600.1581.1612.1699.1862.2132.2311.2483.2637.
 819  2735.2874.2940.2952.2984.2987.2939.2845.2852.2748.2528.2282.
 820  2082.1953.1881.1827.1777.1791.1845.2065.2346.2567.2727.2852.
 820  2942.3051.3078.3081.3097.3100.3022.2906.2897.2739.2514.2231.
 821  2030.1937.1856.1779.1736.1757.1796.1968.2208.2359.2468.2559.
 821  2570.2595.2590.2544.2537.2527.2434.2285.2305.2180.1995.1776.
 822  1618.1529.1487.1437.1428.1461.1521.1703.1900.2062.2170.2253.
 822  2298.2363.2405.2399.2410.2371.2306.2182.2171.2073.1902.1680.
 823  1520.1420.1365.1331.1316.1296.1284.1333.1481.1611.1718.1831.
 823  1871.1909.1956.2013.2061.2041.2026.1930.1940.1847.1701.1543.
 824  1414.1316.1268.1209.1200.1187.1163.1199.1316.1423.1544.1647.
 824  1740.1788.1849.1909.1970.1988.1952.1882.1932.1892.1712.1569.
 825  1496.1402.1385.1352.1343.1416.1493.1667.1905.2101.2260.2355.
 825  2427.2535.2624.2632.2661.2658.2594.2493.2500.2336.2143.1887.
 826  1698.1639.1584.1532.1462.1579.1624.1806.2003.2181.2352.2468.
 826  2590.2713.2805.2825.2859.2867.2780.2637.2668.2532.2292.1998.
 827  1841.1718.1664.1632.1596.1616.1692.1848.2112.2315.2452.2605.
 827  2723.2854.2937.2982.3019.3021.2923.2813.2820.2686.2364.2133.
 828  1914.1851.1771.1696.1687.1698.1799.1959.2163.2350.2465.2601.
 828  2709.2822.2871.2872.2891.2868.2828.2723.2772.2609.2379.2102.
 829  1924.1852.1760.1713.1681.1714.1814.1997.2151.2265.2377.2457.
 829  2552.2586.2626.2587.2570.2517.2435.2350.2348.2216.2052.1844.
 830  1698.1606.1533.1465.1459.1464.1457.1476.1615.1792.1918.2042.
 830  2154.2204.2276.2279.2286.2243.2188.2126.2126.2009.1873.1717.
 831  1603.1520.1453.1397.1358.1360.1346.1376.1527.1651.1756.1849.
 831  1869.1893.1946.1975.2031.2060.2032.2021.2046.2018.1882.1732.
98 0     0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
 9 1  1634.1553.1476.1430.1386.1376.1343.1354.1488.1648.1783.1894.
 9 1  1872.1770.1693.1669.1654.1664.1665.1706.1770.1705.1573.1452.
 9 2  1414.1392.1375.1340.1336.1397.1561.1735.1919.2060.2172.2241.
 9 2  2294.2360.2401.2407.2405.2385.2309.2237.2247.2114.1897.1698.
 9 3  1529.1477.1433.1384.1374.1457.1548.1739.1877.2000.2089.2163.
 9 3  2230.2331.2413.2432.2449.2448.2376.2347.2322.2250.2035.1818.
 9 4  1665.1594.1548.1514.1498.1554.1662.1861.1977.2063.2097.2103.
 9 4  2115.2171.2217.2227.2243.2245.2212.2182.2229.2122.1945.1737.
 9 5  1585.1533.1476.1437.1442.1456.1613.1757.1948.2063.2122.2104.
 9 5  2089.2123.2175.2202.2251.2263.2218.2180.2186.2079.1940.1747.
 9 6  1593.1498.1433.1379.1373.1361.1376.1417.1614.1791.1957.2101.
 9 6  2186.2265.2349.2400.2465.2460.2372.2301.2202.1947.1781.1620.
 9 7  1496.1361.1329.1285.1256.1245.1245.1274.1393.1520.1597.1721.
 9 7  1829.1918.1978.2036.2120.2103.2067.2063.2082.1986.1814.1646.
 9 8  1562.1513.1470.1426.1432.1481.1629.1833.2025.2215.2330.2436.
 9 8  2564.2683.2788.2830.2871.2898.2850.2807.2771.2618.2376.2192.
 9 9  1903.1838.1713.1705.1691.1710.1823.2000.2186.2343.2445.2541.
 9 9  2614.2660.2643.2575.2501.2391.2265.2240.2208.2099.1906.1690.
910  1532.1468.1437.1422.1400.1447.1568.1695.1812.1895.1939.1968.
910  1970.2007.2041.2015.2039.2017.1963.1959.1994.1917.1722.1545.
911  1458.1408.1392.1376.1370.1428.1559.1717.1824.1899.1948.1960.
911  1983.2033.2065.2071.2063.2043.2008.2014.2061.1976.1820.1627.
912  1486.1454.1444.1423.1423.1451.1587.1725.1860.1956.2003.2039.
912  2064.2134.2182.2215.2242.2275.2213.2249.2258.2133.1984.1795.
913  1635.1550.1502.1450.1429.1432.1452.1513.1611.1749.1849.1981.
913  2007.2024.2064.2011.2046.2044.2017.2039.2019.1888.1780.1610.
```

914 1494.1410.1358.1306.1266.1253.1235.1254.1359.1437.1519.1583.
914 1626.1643.1621.1619.1622.1622.1570.1613.1601.1554.1448.1346.
915 1290.1286.1268.1281.1270.1336.1464.1664.1826.1910.1962.1991.
915 1978.2030.2050.2026.2030.2014.1980.2005.1993.1911.1761.1570.
916 1445.1401.1373.1364.1368.1412.1565.1749.1904.1948.1997.2014.
916 1990.2010.2085.2073.2101.2096.2066.2010.1932.1809.1696.1547.
917 1449.1381.1349.1332.1327.1391.1541.1679.1756.1790.1823.1817.
917 1829.1852.1841.1797.1771.1791.1760.1670.1873.1795.1672.1491.
918 1395.1357.1333.1320.1331.1401.1501.1689.1801.1860.1882.1876.
918 1906.1944.1940.1877.1889.1897.1876.1949.1947.1846.1670.1525.
919 1399.1362.1344.1323.1338.1392.1503.1683.1764.1850.1899.1917.
919 1918.1962.1979.1962.1966.1941.1896.1944.1901.1806.1675.1510.
920 1379.1347.1294.1269.1260.1274.1300.1358.1468.1599.1704.1798.
920 1849.1928.1967.2024.2056.2053.2004.2005.1975.1880.1764.1612.
921 1486.1420.1368.1339.1310.1305.1307.1328.1473.1619.1723.1840.
921 1941.2010.2055.2086.2139.2137.2101.2132.2141.2038.1876.1717.
922 1638.1599.1566.1545.1538.1572.1734.1931.2164.2366.2450.2565.
922 2527.2586.2573.2430.2255.2178.2132.2109.2062.1939.1759.1628.
923 1445.1410.1375.1339.1359.1409.1538.1681.1806.1854.1892.1898.
923 1880.1910.1887.1837.1828.1815.1820.1902.1870.1793.1678.1505.
924 1418.1419.1382.1343.1383.1424.1582.1737.1845.1908.1944.1934.
924 1930.1960.1934.1900.1904.1887.1902.1992.1952.1873.1733.1530.
925 1453.1425.1403.1378.1383.1449.1588.1744.1864.1950.1984.2015.
925 2019.2047.2010.1969.1934.1902.1865.1966.1958.1865.1713.1533.
926 1394.1382.1350.1332.1345.1408.1579.1702.1825.1880.1904.1874.
926 1850.1877.1853.1787.1760.1769.1775.1854.1810.1728.1629.1458.
927 1353.1310.1283.1263.1242.1254.1305.1335.1476.1535.1564.1574.
927 1539.1524.1505.1479.1498.1520.1535.1602.1580.1503.1422.1302.
928 1228.1192.1163.1143.1132.1142.1156.1190.1267.1327.1353.1388.
928 1399.1403.1397.1366.1402.1412.1423.1539.1520.1464.1378.1275.
929 1260.1249.1253.1256.1271.1309.1465.1647.1780.1870.1897.1918.
929 1893.1942.1936.1889.1886.1873.1864.1943.1907.1806.1662.1498.
930 1391.1340.1327.1313.1336.1401.1531.1687.1794.1884.1943.1976.
930 1992.2021.2027.1981.1992.1984.1955.2023.1980.1880.1735.1556.
98 0 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
10 1 1418.1384.1379.1348.1351.1397.1541.1733.1872.1954.2001.2026.
10 1 2041.2098.2092.2056.2024.2014.2028.2044.1986.1849.1683.1505.
10 2 1404.1357.1346.1333.1335.1387.1524.1706.1801.1862.1888.1895.
10 2 1879.1892.1913.1838.1824.1826.1849.1916.1887.1807.1683.1503.
10 3 1424.1372.1363.1337.1355.1405.1565.1745.1868.1914.1929.1913.
10 3 1882.1881.1892.1814.1774.1795.1828.1680.1820.1758.1642.1488.
10 4 1382.1338.1303.1288.1271.1288.1362.1421.1534.1593.1621.1597.
10 4 1542.1537.1505.1499.1507.1534.1571.1637.1595.1550.1470.1368.
10 5 1301.1241.1237.1232.1230.1268.1305.1330.1390.1411.1407.1426.
10 5 1425.1404.1372.1346.1382.1372.1440.1541.1524.1495.1402.1318.
10 6 1288.1283.1284.1275.1302.1391.1556.1737.1854.1906.1934.1925.
10 6 1884.1882.1850.1802.1794.1791.1829.1929.1897.1780.1687.1499.
10 7 1407.1356.1333.1330.1357.1418.1596.1746.1878.1938.1927.1933.
10 7 1913.1937.1925.1874.1855.1845.1882.1975.1922.1757.1642.1506.
10 8 1379.1368.1370.1341.1377.1429.1586.1758.1829.1884.1916.1943.
10 8 1948.2004.1988.2010.2023.2001.2006.2068.2026.1897.1762.1576.
10 9 1445.1385.1380.1330.1349.1427.1567.1737.1844.1907.1926.1943.
10 9 1928.2001.1988.1958.1940.1934.1942.2001.1959.1872.1725.1521.
1010 1437.1389.1345.1348.1367.1428.1615.1740.1830.1904.1958.1962.
1010 1948.1955.1942.1893.1843.1821.1829.1899.1834.1760.1643.1486.
1011 1366.1317.1283.1266.1263.1287.1335.1373.1472.1537.1566.1569.
1011 1541.1534.1503.1496.1510.1517.1557.1587.1559.1500.1424.1321.
1012 1247.1212.1195.1191.1198.1223.1264.1280.1346.1375.1379.1389.
1012 1389.1357.1311.1316.1337.1340.1426.1510.1481.1464.1390.1333.
1013 1295.1300.1216.1305.1319.1378.1536.1668.1818.1860.1886.1878.
1013 1842.1822.1789.1761.1758.1809.1884.1921.1859.1785.1652.1490.

```
1014  1398.1375.1348.1332.1351.1418.1548.1720.1866.1882.1891.1898.
1014  1869.1900.1837.1814.1844.1822.1879.1925.1865.1797.1673.1522.
1015  1391.1360.1338.1328.1350.1399.1557.1721.1794.1868.1883.1877.
1015  1858.1883.1880.1822.1812.1837.1923.1916.1874.1795.1684.1473.
1016  1359.1339.1320.1298.1319.1377.1551.1730.1818.1876.1891.1883.
1016  1892.1928.1908.1878.1853.1874.1925.1931.1883.1805.1693.1516.
1017  1394.1357.1335.1314.1363.1379.1510.1721.1792.1872.1877.1878.
1017  1839.1838.1819.1779.1767.1755.1824.1849.1783.1708.1593.1449.
1018  1318.1285.1257.1236.1224.1239.1312.1364.1474.1522.1546.1544.
1018  1494.1476.1465.1420.1446.1463.1542.1571.1528.1470.1404.1311.
1019  1240.1186.1171.1153.1152.1162.1210.1244.1317.1353.1375.1383.
1019  1385.1375.1334.1321.1332.1351.1465.1527.1495.1468.1397.1295.
1020  1276.1259.1262.1251.1276.1352.1540.1736.1841.1872.1885.1876.
1020  1832.1842.1805.1749.1743.1764.1863.1889.1832.1776.1647.1494.
1021  1398.1354.1347.1332.1353.1421.1619.1758.1845.1854.1873.1853.
1021  1833.1875.1850.1798.1789.1786.1891.1911.1861.1802.1678.1492.
1022  1390.1353.1334.1326.1340.1398.1591.1777.1839.1855.1876.1846.
1022  1833.1842.1817.1760.1741.1750.1859.1872.1816.1766.1631.1477.
1023  1378.1338.1329.1315.1334.1398.1564.1719.1817.1852.1866.1850.
1023  1817.1840.1828.1748.1741.1761.1857.1869.1810.1723.1608.1462.
1024  1350.1329.1293.1291.1282.1353.1528.1719.1839.1893.1932.1911.
1024  1882.1871.1848.1803.1798.1825.1893.1862.1810.1733.1626.1457.
1025  1378.1343.1318.1298.1307.1314.1398.1442.1549.1605.1611.1602.
1025  1559.1531.1520.1515.1522.1574.1655.1642.1608.1553.1488.1391.
1026  1275.1267.1250.1239.1239.1279.1288.1354.1417.1425.1427.1413.
1026  1406.1374.1344.1328.1395.1548.1579.1560.1529.1488.1399.1273.
1027  1295.1298.1311.1303.1331.1415.1600.1781.1906.2001.2011.2015.
1027  1976.1973.1973.1939.1969.2054.2035.1963.1879.1793.1615.1503.
1028  1428.1426.1405.1430.1447.1525.1676.1866.1929.1999.1997.1989.
1028  1951.1947.1905.1863.1909.2072.2089.2024.1998.1887.1776.1621.
1029  1549.1547.1515.1520.1537.1632.1742.1904.1984.1973.1951.1926.
1029  1880.1903.1878.1802.1831.2009.2027.1987.1951.1860.1742.1621.
1030  1538.1522.1486.1486.1520.1598.1723.1879.1951.1941.1942.1899.
1030  1821.1822.1796.1748.1818.1975.1979.1947.1912.1840.1732.1568.
1031  1490.1423.1465.1457.1477.1556.1661.1812.1884.1897.1892.1855.
1031  1809.1775.1766.1733.1745.1875.1862.1788.1744.1684.1593.1436.
98 0     0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
11 1  1351.1304.1291.1285.1292.1317.1334.1393.1478.1535.1528.1522.
11 1  1464.1449.1421.1416.1452.1656.1618.1590.1536.1485.1404.1322.
11 2  1265.1229.1195.1216.1212.1230.1244.1284.1348.1365.1365.1364.
11 2  1359.1323.1297.1290.1362.1516.1541.1500.1489.1429.1358.1277.
11 3  1246.1224.1242.1258.1264.1337.1494.1676.1785.1821.1841.1814.
11 3  1770.1800.1760.1739.1749.1892.1893.1844.1798.1697.1575.1412.
11 4  1370.1317.1341.1331.1360.1440.1581.1717.1842.1881.1893.1876.
11 4  1850.1845.1805.1758.1774.1955.1962.1885.1871.1807.1686.1535.
11 5  1432.1425.1385.1410.1416.1519.1659.1805.1883.1925.1912.1902.
11 5  1861.1863.1832.1783.1801.1968.1998.1941.1904.1849.1725.1578.
11 6  1494.1467.1459.1443.1464.1547.1684.1847.1901.1909.1884.1879.
11 6  1853.1856.1846.1769.1801.1974.1984.1918.1856.1794.1663.1508.
11 7  1424.1396.1379.1371.1377.1439.1590.1754.1843.1878.1892.1880.
11 7  1850.1862.1844.1804.1762.1916.1912.1840.1785.1677.1603.1447.
11 8  1319.1277.1269.1250.1249.1295.1302.1375.1471.1519.1518.1462.
11 8  1465.1433.1416.1406.1462.1617.1604.1539.1480.1413.1339.1245.
11 9  1174.1144.1114.1116.1095.1125.1133.1162.1244.1261.1295.1317.
11 9  1332.1308.1255.1290.1313.1497.1499.1485.1449.1417.1347.1253.
1110  1244.1238.1248.1261.1264.1350.1516.1677.1793.1842.1845.1837.
1110  1790.1607.1769.1731.1786.1980.1966.1930.1864.1835.1720.1561.
1111  1451.1430.1431.1405.1454.1475.1548.1650.1761.1799.1797.1756.
1111  1747.1711.1693.1680.1747.1898.1917.1859.1828.1772.1665.1508.
1112  1469.1465.1465.1446.1472.1539.1694.1840.1922.1946.1929.1920.
1112  1857.1862.1853.1796.1894.2016.2028.1978.1897.1846.1764.1593.
```

```
1113  1512.1478.1447.1457.1457.1538.1693.1838.1906.1912.1934.1928.
1113  1877.1882.1844.1822.1853.2004.1976.1909.1836.1794.1673.1508.
1114  1418.1403.1385.1383.1400.1418.1610.1788.1884.1939.1988.1994.
1114  1932.1943.1922.1872.1929.2034.2003.1941.1867.1802.1701.1545.
1115  1448.1394.1363.1360.1375.1380.1424.1481.1587.1667.1682.1677.
1115  1656.1634.1615.1584.1669.1782.1734.1689.1645.1580.1521.1438.
1116  1369.1324.1298.1305.1286.1302.1315.1355.1396.1422.1420.1452.
1116  1455.1435.1424.1440.1537.1656.1658.1627.1619.1566.1480.1405.
1117  1394.1388.1395.1395.1437.1501.1706.1873.2006.2036.2055.2037.
1117  2003.2011.1976.1949.2036.2166.2141.2066.2025.1950.1802.1644.
1118  1558.1543.1531.1504.1511.1590.1746.1908.2022.2023.2026.2004.
1118  1972.1973.1941.1907.1977.2126.2110.2038.2003.1946.1853.1644.
1119  1566.1591.1550.1560.1578.1628.1794.1937.2009.2020.1993.1967.
1119  1925.1920.1900.1849.1922.2134.2136.2057.2023.1957.1838.1676.
1120  1590.1549.1559.1573.1564.1636.1815.1930.2019.2008.1979.1971.
1120  1917.1916.1866.1832.1906.2103.2057.2009.1974.1919.1803.1625.
1121  1524.1509.1510.1518.1546.1632.1776.1893.1970.1999.1980.1947.
1121  1878.1856.1799.1765.1812.2001.1990.1941.1896.1844.1759.1597.
1122  1481.1453.1435.1412.1430.1455.1508.1539.1623.1708.1683.1661.
1122  1630.1633.1584.1596.1690.1814.1794.1726.1672.1624.1531.1393.
1123  1318.1281.1265.1251.1265.1272.1311.1339.1418.1478.1503.1549.
1123  1555.1532.1508.1502.1605.1687.1663.1630.1609.1565.1475.1388.
1124  1359.1362.1371.1356.1400.1463.1663.1821.2002.2066.2094.2081.
1124  2002.2028.1994.1959.2016.2182.2136.2089.2052.1977.1847.1884.
1125  1587.1586.1565.1575.1603.1665.1827.1950.2034.2046.2033.2005.
1125  1960.1946.1930.1896.1976.2146.2121.2073.2045.2011.1875.1750.
1126  1663.1634.1586.1605.1618.1702.1834.2006.2110.2128.2112.2086.
1126  2035.2097.2024.2039.2087.2171.2138.2067.2041.1952.1870.1653.
1127  1501.1448.1441.1426.1420.1427.1481.1510.1587.1613.1620.1585.
1127  1507.1443.1401.1402.1465.1540.1560.1547.1527.1506.1440.1354.
1128  1301.1278.1262.1263.1271.1295.1389.1479.1594.1662.1652.1640.
1128  1573.1585.1577.1599.1687.1822.1792.1717.1699.1627.1542.1440.
1129  1392.1352.1324.1340.1322.1375.1423.1444.1537.1579.1591.1600.
1129  1536.1504.1473.1448.1560.1746.1709.1667.1628.1574.1500.1406.
1130  1324.1274.1272.1262.1257.1292.1317.1311.1402.1412.1426.1430.
1130  1429.1387.1367.1384.1492.1651.1619.1581.1542.1517.1420.1330.
98 0     0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
12 1  1292.1305.1307.1311.1336.1418.1617.1788.1896.1925.1948.1939.
12 1  1927.1928.1924.1918.1973.2094.2042.1977.1929.1871.1769.1576.
12 2  1465.1451.1462.1454.1518.1599.1840.2005.2105.2178.2200.2182.
12 2  2146.2151.2118.2095.2171.2300.2272.2251.2206.2126.1990.1802.
12 3  1714.1675.1668.1673.1684.1769.1945.2090.2172.2210.2191.2153.
12 3  2072.2054.2076.2017.2099.2267.2234.2218.2175.2106.1950.1754.
12 4  1674.1644.1630.1601.1599.1664.1843.2004.2083.2120.2126.2104.
12 4  2068.2070.2054.2047.2123.2203.2142.2116.2036.1978.1833.1654.
12 5  1535.1493.1487.1449.1465.1541.1690.1863.1936.1997.1966.1954.
12 5  1911.1897.1851.1841.1906.2019.1998.1924.1859.1799.1697.1518.
12 6  1409.1355.1341.1313.1297.1311.1376.1408.1515.1602.1591.1584.
12 6  1554.1523.1514.1503.1596.1741.1698.1669.1617.1557.1469.1348.
12 7  1249.1189.1161.1148.1134.1159.1199.1229.1288.1350.1367.1406.
12 7  1422.1425.1411.1406.1517.1608.1598.1559.1538.1493.1422.1318.
12 8  1286.1289.1268.1278.1283.1350.1539.1783.1899.1948.1999.1988.
12 8  1956.1981.1979.1951.2034.2120.2111.2052.2018.1954.1839.1652.
12 9  1536.1491.1499.1483.1498.1559.1742.1943.2038.2075.2123.2108.
12 9  2090.2074.2079.2032.2121.2231.2215.2145.2094.2000.1820.1730.
1210  1630.1612.1599.1597.1614.1687.1848.2026.2104.2147.2153.2150.
1210  2110.2148.2113.2122.2187.2311.2245.2220.2172.2080.1935.1775.
1211  1664.1609.1605.1602.1632.1686.1859.2011.2080.2129.2135.2111.
1211  2051.2057.2015.1996.2060.2198.2174.2126.2071.2009.1889.1718.
1212  1592.1565.1551.1553.1562.1659.1791.1939.1991.2032.2006.1942.
1212  1868.1881.1826.1802.1858.2024.1994.1928.1687.1793.1722.1568.
```

```
1213                1435.1373.1381.1341.1346.1378.1481.1544.1634.1702.1694.1714.
1213                1687.1648.1620.1614.1715.1886.1866.1817.1788.1737.1656.1568.
1214                1449.1392.1352.1359.1351.1369.1393.1434.1488.1495.1504.1506.
1214                1508.1468.1458.1451.1551.1714.1726.1682.1669.1631.1533.1428.
1215                1400.1376.1377.1380.1400.1504.1682.1864.1960.1989.2026.2008.
1215                1969.1960.1918.1899.1980.2151.2127.2083.2047.1981.1859.1656.
1216                1550.1528.1519.1500.1519.1580.1772.1939.2054.2111.2090.2079.
1216                2031.2006.1980.1911.2048.2211.2203.2135.2112.2035.1907.1720.
1217                1641.1612.1607.1605.1622.1695.1838.1976.2049.2087.2097.2045.
1217                1991.1992.1977.1947.2028.2203.2201.2135.2084.2009.1874.1690.
1218                1571.1571.1547.1553.1551.1643.1838.1986.2053.2073.2071.2070.
1218                2025.2035.2017.2018.2122.2253.2242.2167.2151.2084.1955.1793.
1219                1694.1683.1690.1705.1723.1811.1990.2158.2238.2271.2288.2248.
1219                2185.2137.2133.2125.2197.2362.2346.2302.2231.2184.2079.1916.
1220                1788.1752.1741.1729.1726.1747.1817.1843.1935.1972.1967.1942.
1220                1877.1838.1819.1806.1901.2087.2076.2031.1984.1938.1846.1719.
1221                1622.1573.1562.1518.1542.1561.1601.1632.1697.1724.1718.1691.
1221                1689.1660.1630.1618.1737.1891.1911.1878.1879.1830.1762.1660.
1222                1640.1637.1632.1631.1645.1710.1871.2027.2117.2168.2164.2154.
1222                2078.2038.2019.1971.2074.2224.2193.2159.2119.2060.1927.1751.
1223                1649.1593.1586.1590.1587.1662.1827.1967.2047.2100.2077.2057.
1223                1991.1906.1931.1883.1975.2120.2104.2045.2000.1908.1793.1639.
1224                1456.1414.1386.1394.1391.1467.1554.1641.1770.1826.1849.1842.
1224                1783.1760.1724.1709.1800.1888.1813.1764.1730.1721.1677.1602.
1225                1537.1494.1463.1473.1464.1502.1549.1578.1623.1620.1594.1557.
1225                1499.1463.1443.1464.1545.1661.1674.1667.1666.1630.1569.1493.
1226                1443.1413.1393.1401.1402.1413.1501.1574.1668.1723.1730.1722.
1226                1695.1663.1644.1652.1713.1833.1848.1799.1746.1683.1592.1523.
1227                1455.1420.1402.1386.1380.1416.1472.1500.1600.1679.1704.1690.
1227                1633.1615.1590.1585.1677.1822.1825.1773.1743.1681.1580.1510.
1228                1434.1394.1371.1351.1365.1368.1413.1433.1476.1492.1488.1502.
1228                1482.1442.1428.1431.1541.1661.1657.1627.1611.1582.1497.1410.
1229                1372.1344.1342.1353.1363.1397.1507.1664.1799.1919.1981.1982.
1229                1959.1950.1916.1896.1936.2037.2008.1942.1906.1809.1720.1579.
1230                1500.1466.1441.1436.1435.1490.1604.1722.1799.1894.1934.1924.
1230                1883.1908.1882.1860.1921.2031.2027.1972.1942.1856.1766.1589.
1231                1477.1462.1444.1421.1425.1473.1577.1632.1738.1774.1801.1799.
1231                1767.1756.1733.1707.1729.1836.1824.1729.1622.1559.1525.1461.
99
UBAS
UBAS PLANT A  1    3   0.00   .742
UBAS PLANT A  2    3   0.00   .742
UBAS PLANT A  3    3   0.00   .742
UBAS PLANT B  1   10   0.00   .898
UBAS PLANT B  2   10   0.00   .898
UBAS PLANT B  3   10   0.00   .898
UBAS PLANT B  4    6   0.00  1.133
UBAS PLANT B  5    6   0.00  1.410
UBAS PLANT C  1    4   0.00   .598
UBAS PLANT C  2    4   0.00   .598
UBAS PLANT D  1    5   0.00   .868
UBAS PLANT D  2    5   0.00   .868
UBAS PLANT E  1    9   0.00   .725
UBAS PLANT E  2    9   0.00   .725
UBAS PLANT E  3    9   0.00   .725
UBAS PLANT E  4    9   0.00   .725
UBAS PLANT E  5    9   0.00   .725
UBAS PLANT E  6    7   0.00   .951
UBAS PLANT F  1   11   0.00  2.500
UBAS PLANT F  2   11   0.00  2.500
UBAS PLANT F  3   11   0.00  2.500
```

```
UBAS PLANT F  4   11    0.00   2.500
UBAS PLANT G  1   11    0.00   2.500
UBAS PLANT G  2   11    0.00   2.500
UBAS PLANT G  3   11    0.00   2.500
UBAS PLANT G  4   11    0.00   2.500
UBAS PLANT H  3   12    0.00   2.500
UBAS PLANT I  1   12    0.00   2.500
UBAS PLANT J  1   12    0.00   2.500
UBAS PLANT K  1    2   65.93   2.618
UBAS PLANT K  2    2   52.60   2.032
UBAS PLANT K  3    2   64.29   2.032
UBAS PLANT K  4    2   64.29   2.032
UBAS PURCHASE 1   12    0.00   2.500
UCAP
UCAP PLANT A  1 0  280.   420.   575.   0.   0.   0.   0.
UCAP PLANT A  2 0  280.   420.   575.   0.   0.   0.   0.
UCAP PLANT A  3 1  280.   420.   575.   0.   0.   0.   0.
UCAP PLANT B  1 0   10.    30.    47.   0.   0.   0.   0.
UCAP PLANT B  2 0   10.    30.    47.   0.   0.   0.   0.
UCAP PLANT B  3 0   10.    30.    47.   0.   0.   0.   0.
UCAP PLANT B  4 0   25.    60.   100.   0.   0.   0.   0.
UCAP PLANT B  5 0  120.   240.   360.   0.   0.   0.   0.
UCAP PLANT C  1 0   22.    50.    75.   0.   0.   0.   0.
UCAP PLANT C  2 0   75.   155.   220.   0.   0.   0.   0.
UCAP PLANT D  1 0   22.    50.    75.   0.   0.   0.   0.
UCAP PLANT D  2 1   25.    65.   102.   0.   0.   0.   0.
UCAP PLANT E  1 0    6.    28.    48.   0.   0.   0.   0.
UCAP PLANT E  2 0    6.    28.    48.   0.   0.   0.   0.
UCAP PLANT E  3 0    6.    28.    48.   0.   0.   0.   0.
UCAP PLANT E  4 0    6.    28.    48.   0.   0.   0.   0.
UCAP PLANT E  5 1    6.    28.    48.   0.   0.   0.   0.
UCAP PLANT E  6 0  126.   275.   425.   0.   0.   0.   0.
UCAP PLANT F  1 0   20.     0.     0.   0.   0.   0.   0.
UCAP PLANT F  2 0   20.     0.     0.   0.   0.   0.   0.
UCAP PLANT F  3 0   20.     0.     0.   0.   0.   0.   0.
UCAP PLANT F  4 0   20.     0.     0.   0.   0.   0.   0.
UCAP PLANT G  1 0   14.     0.     0.   0.   0.   0.   0.
UCAP PLANT G  2 0   14.     0.     0.   0.   0.   0.   0.
UCAP PLANT G  3 0   14.     0.     0.   0.   0.   0.   0.
UCAP PLANT G  4 0   14.     0.     0.   0.   0.   0.   0.
UCAP PLANT H  3 0   11.     0.     0.   0.   0.   0.   0.
UCAP PLANT I  1 0   12.     0.     0.   0.   0.   0.   0.
UCAP PLANT J  1 0   11.     0.     0.   0.   0.   0.   0.
UCAP PLANT K  1 0  113.   270.   450.   0.   0.   0.   0.
UCAP PLANT K  2 0  113.   270.   450.   0.   0.   0.   0.
UCAP PLANT K  3 0  150.   330.   550.   0.   0.   0.   0.
UCAP PLANT K  4 0  150.   330.   550.   0.   0.   0.   0.
UCAP PURCHASE 1 1  100.     0.     0.   0.   0.   0.   0.
UFOR
UFOR PLANT A  1   10.2
UFOR PLANT A  2    5.2
UFOR PLANT A  3    9.8
UFOR PLANT B  1    2.3
UFOR PLANT B  2    2.3
UFOR PLANT B  3    2.3
UFOR PLANT B  4    8.3
UFOR PLANT B  5    6.9
UFOR PLANT C  1    4.3
UFOR PLANT C  2    3.3
UFOR PLANT D  1    1.0
UFOR PLANT D  2    2.4
```

```
UFOR PLANT E  1     1.5
UFOR PLANT E  2     1.5
UFOR PLANT E  3     1.5
UFOR PLANT E  4     1.5
UFOR PLANT E  5     1.5
UFOR PLANT E  6     9.8
UFOR PLANT F  1    15.8
UFOR PLANT F  2    15.8
UFOR PLANT F  3    15.8
UFOR PLANT F  4    15.8
UFOR PLANT G  1    15.8
UFOR PLANT G  2    15.8
UFOR PLANT G  3    15.8
UFOR PLANT G  4    15.8
UFOR PLANT H  3    15.8
UFOR PLANT I  1    15.8
UFOR PLANT J  1    15.8
UFOR PLANT K  1    10.4
UFOR PLANT K  2    10.4
UFOR PLANT K  3    12.1
UFOR PLANT K  4    12.1
UFOR PURCHASE 1    25.0
UHRT
UHRT PLANT A  1  Q  670.59   7.9580   .00116
UHRT PLANT A  2  Q  348.62   8.6306   .00093
UHRT PLANT A  3  Q  127.68  10.2360  -.00150
UHRT PLANT B  1  Q  214.56  10.0000  0.00000
UHRT PLANT B  2  Q  214.56  10.0000  0.00000
UHRT PLANT B  3  Q  214.56  10.0000  0.00000
UHRT PLANT B  4  Q  133.12   8.5223   .00683
UHRT PLANT B  5  Q  503.00   6.2518   .00563
UHRT PLANT C  1  Q  103.40   9.4365   .00247
UHRT PLANT C  2  Q  279.00   7.0068   .00498
UHRT PLANT D  1  Q   86.60   9.4271   .00368
UHRT PLANT D  2  Q  143.01   7.6613   .01467
UHRT PLANT E  1  Q  102.15   5.5735   .09508
UHRT PLANT E  2  Q  102.15   5.5735   .09508
UHRT PLANT E  3  Q  102.15   5.5735   .09508
UHRT PLANT E  4  Q  102.15   5.5735   .09508
UHRT PLANT E  5  Q  102.15   5.5735   .09508
UHRT PLANT E  6  Q  526.53   9.3345  0.00000
UHRT PLANT F  1     16750.      0.      0.    0.    0.    0.    0.
UHRT PLANT F  2     16750.      0.      0.    0.    0.    0.    0.
UHRT PLANT F  3     16750.      0.      0.    0.    0.    0.    0.
UHRT PLANT F  4     16750.      0.      0.    0.    0.    0.    0.
UHRT PLANT G  1     16600.      0.      0.    0.    0.    0.    0.
UHRT PLANT G  2     16600.      0.      0.    0.    0.    0.    0.
UHRT PLANT G  3     16600.      0.      0.    0.    0.    0.    0.
UHRT PLANT G  4     16600.      0.      0.    0.    0.    0.    0.
UHRT PLANT H  3     17300.      0.      0.    0.    0.    0.    0.
UHRT PLANT I  1     33500.      0.      0.    0.    0.    0.    0.
UHRT PLANT J  1     11000.      0.      0.    0.    0.    0.    0.
UHRT PLANT K  1      8681.   8681.   8681.    0.    0.    0.    0.
UHRT PLANT K  2      8681.   8681.   8681.    0.    0.    0.    0.
UHRT PLANT K  3      8649.   8649.   8649.    0.    0.    0.    0.
UHRT PLANT K  4      8649.   8649.   8649.    0.    0.    0.    0.
UHRT PURCHASE 1     36850.      0.      0.    0.    0.    0.    0.
LAST
```

APPENDIX C

COMPUTER SIMULATION CODE AND RESULTS

The FORTRAN computer codes for programs PREXPAN and XPAN are listed as follows

```
C##############################################################################C
C##############################################################################C
C                                                                              C
C              PROGRAM PREXPAN              BY ERIC KRULL                      C
C                                                                              C
C##############################################################################C
C##############################################################################C


      PROGRAM PREXPAN(INPUT,OUTPUT,DATIN,DATOUT,MAX,AVE,COSDAT,MAXPTS,
C
C  PROGRAM PREXPAN SETS UP THE FILES FOR PROGRAM XPAN AND ALSO
C  CALCULATES THE PROBABILITY DISTRIBUTION CURVES FOR THE MAXIMUM
C  AND AVERAGE LOAD WEEKS.
C
     +AVEPTS,XFUEL,XFCST,XLLIB,XUBAS,XUCAP,XUFOR,XUHRT,
     +TAPE3=MAX,TAPE4=AVE,TAPE5=COSDAT,TAPE6=OUTPUT,
     +TAPE10=MAXPTS,TAPE11=AVEPTS,TAPE21=XFUEL,TAPE22=XFCST,TAPE23=
     +XLLIB,TAPE24=XUBAS,TAPE25=XUCAP,TAPE26=XUFOR,TAPE27=XUHRT,
     +TAPE1=DATIN,TAPE7=INPUT,TAPE2=DATOUT)
      DIMENSION LDPEAK(200,2)
      REAL AVPEAK(7,24),PEAKWK1(7,24),PEAKWK2(7,24),LOAD(24)
      REWIND 1
      READ(1,800)IDAYONE,IDAY365
      CALL GENDATA
      CALL LLOAD(IDAYONE,IDAY365)
      CALL LDDIST
  800 FORMAT(17X,2(1X,I1))
      STOP
      END


C##############################################################################C
C##############################################################################C
C                                                                              C
C                     SUBROUTINE GENDATA                                       C
C                                                                              C
C##############################################################################C
C##############################################################################C


      SUBROUTINE GENDATA
C
C  SUBROUTINE GENDATA SORTS THE INPUT GENERATOR DATA FROM THE COSDAT
C  FILE AND PUTS IT INTO SEPERATE FILES FOR USE BY OTHER SUBROUTINES
C
      INTEGER X,LAST,Q,UCAP,UFOR,UHRT,FUEL,FCST,UBAS
      REWIND 5
      REWIND 21
      REWIND 22
      REWIND 23
      REWIND 24
      REWIND 25
      REWIND 26
      REWIND 27
      FUEL="FUEL"
      FCST="FCST"
      LAST="LAST"
      LLIB="LLIB"
      UCAP="UCAP"
      UFOR="UFOR"
```

```
      UHRT="UHRT"
      UBAS="UBAS"
      Q="Q"
6000  READ(5,800)X
6008  IF(X.EQ.LAST)GOTO6009
      IF(X.EQ.FUEL)ASSIGN 6001 TO INUM
      IF(X.EQ.FCST)ASSIGN 6002 TO INUM
      IF(X.EQ.LLIB)ASSIGN 6003 TO INUM
      IF(X.EQ.UBAS)ASSIGN 6004 TO INUM
      IF(X.EQ.UCAP)ASSIGN 6005 TO INUM
      IF(X.EQ.UFOR)ASSIGN 6006 TO INUM
      IF(X.EQ.UHRT)ASSIGN 6007 TO INUM
      IF(X.NE.FUEL.AND.X.NE.FCST.AND.X.NE.LLIB.AND.X.NE.UBAS.AND.X.NE.
     +UCAP.AND.X.NE.UFOR.AND.X.NE.UHRT)GOTO6000
      GOTO INUM,(6001,6002,6003,6004,6005,6006,6007)
6001  READ(5,801)X,NUMBER,NAME,UNITS,HEATRT,ESCALN
      IF(X.NE.FUEL)GOTO6008
      WRITE(21,901)NUMBER,NAME,UNITS,HEATRT,ESCALN
      GOTO6001
6002  READ(5,802)X,IYEAR,NUMBER,C1
      IF(X.NE.FCST)GOTO60C8
      WRITE(22,902)NUMBER,IYEAR,C1
      GO TO 6002
6003  READ(5,803)MONTH,DAY,I1,I2,I3,I4,I5,I6,I7,I8,I9,I10,I11,I12
      IF(MONTH.EQ.99)GOTO6000
      WRITE(23,903)MONTH,DAY,I1,I2,I3,I4,I5,I6,I7,I8,I9,I10,I11,I12
      GOTO6003
6004  READ(5,804)X,NAME,NUMBER,LFUEL,FXOPCOS,VARCOST
      IF(X.NE.UBAS)GOTO6008
      WRITE(24,904)NAME,NUMBER,LFUEL,FXOPCOS,VARCOST
      GO TO 6004
6005  READ(5,805)X,NAME,NUMBER,II,CAP1,CAP2,CAP3,CAP4,CAP5,CAP6,CAP7
      IF(X.NE.UCAP)GOTO6008
      WRITE(25,905)NAME,NUMBER,II,CAP1,CAP2,CAP3,CAP4,CAP5,CAP6,CAP7
      GOTO6005
6006  READ(5,806)X,NAME,NUMBER,FORATE
      IF(X.NE.UFOR)GOTO6008
      WRITE(26,906)NAME,NUMBER,FORATE
      GOTO6006
6007  READ(5,807)X,NAME,NUMBER,XX,RT1,RT2,RT3,RT4,RT5,RT6,RT7
      IF(X.NE.UHRT)GOTO6008
      IF(XX.NE.Q)WRITE(27,907)NAME,NUMBER,XX,RT1,RT2,RT3,RT4,RT5,RT6,RT7
      IF(XX.EQ.Q)WRITE(27,908)NAME,NUMBER,XX,RT1,RT2,RT3
      GOTO6007
6009      DO 105 IFILE=21,27
          IF(IFILE.LE.22)WRITE(IFILE,910)
 105      IF(IFILE.GT.22)WRITE(IFILE,909)
      RETURN
 800  FORMAT(A4)
 801  FORMAT(A4,1X,I2,2X,A8,2X,A4,2X,F9.2,2X,F5.1)
 802  FORMAT(A4,1X,2(I2,2X),3X,F7.1)
 803  FORMAT(I2,I2,16X,12F5.0)
 804  FORMAT(A4,1X,A8,I2,2X,I3,2X,F5.2,2X,F5.3)
 805  FORMAT(A4,1X,A8,I2,1X,I1,F6.0,6(2X,F6.0))
 806  FORMAT(A4,1X,A8,I2,2X,F5.1)
 807  FORMAT(A4,1X,A8,I2,2X,A1,2X,F6.0,2(2X,F7.0),4(2X,F6.0))
 901  FORMAT(I2,2X,A8,2X,A4,2X,F9.2,2X,F5.1)
 902  FORMAT(2(I2,2X),3X,F7.1)
 903  FORMAT(I2,I2,16X,12F5.0)
 904  FORMAT(A8,I2,2X,I3,2X,F5.2,2X,F5.3)
 905  FORMAT(A8,I2,1X,I1,F6.0,6(2X,F6.0))
```

```
123    906 FORMAT(A8,I2,2X,F5.1)
124    907 FORMAT(A8,I2,2X,A1,2X,F6.0,2(2X,F7.0),4(2X,F6.0))
125    908 FORMAT(A8,I2,2X,A1,2X,F6.2,2X,F7.4,2X,F7.5)
126    909 FORMAT("LAST")
127    910 FORMAT(2H99)
128        END
129
130
131  C##########################################################################C
132  C##########################################################################C
133  C                                                                          C
134  C                          SUBROUTINE LLOAD                                C
135  C                                                                          C
136  C##########################################################################C
137  C##########################################################################C
138
139
140        SUBROUTINE LLOAD(IDAYONE,IDAY365)
141  C
142  C   SUBROUTINE LOAD CALCULATES THE LOAD PROFILE CURVE FOR THE AVERAGE
143  C   LOAD WEEK AND FINDS THE MAXIMUM WEEK LOAD PROFILE CURVE BY SEARCHING
144  C   AND COMPARISON.
145  C
146        REAL AVPEAK(7,24),PEAKWK1(7,24),PEAKWK2(7,24),LOAD(24)
147        INTEGER MONTH,DAY1,DAY2,DAY
148        REWIND 3
149        REWIND 4
150        REWIND 23
151        IIDAY=WKSUM1=WKSUM2=0
152           DO 5 IX=1,7
153                DO 15 IY=1,24
154                AVPEAK(IX,IY)=0.
155     15         PEAKWK2(IX,IY)=0.
156      5     CONTINUE
157            DO 105 IWEEK=1,52
158                 DO 115 IDAY=1,7
159  C
160  C   SET UP TWO LOOPS THAT WILL READ IN LOAD DATA FOR THE 52 WEEKS OF THE
161  C   YEAR, 7 DAYS A WEEK.  IDAYONE HOLDS THE VALUE OF THE DAY OF THE WEEK
162  C   FOR JAN. 1.
163  C
164                 IF(IWEEK.EQ.1.AND.IDAY.LT.IDAYONE)GOTO115
165      2          READ(23,820)MONTH,DAY,(LOAD(I),I=1,12)
166  C
167  C   READ IN THE DAY'S FIRST 12 HOURS LOAD DATA
168  C
169                 IF(MONTH.EQ.98)GO TO 2
170                 READ(23,820)MONTH,DAY,(LOAD(I),I=13,24)
171  C
172  C   READ IN THE DAY'S SECOND 12 HOURS LOAD DATA
173  C
174                 IF(IWEEK.EQ.1)GOTO3
175                 IIDAY=IIDAY+1
176      3               DO 125 IHOUR=1,24
177                      AVPEAK(IDAY,IHOUR)=AVPEAK(IDAY,IHOUR)+LOAD(IHOUR)
178  C
179  C   AVPEAK HOLDS THE HOURLY LOAD SUMS FOR THE ENTIRE WEEK.  AVPEAK WILL
180  C   BE USED LATER TO FIND THE AVERAGE HOURLY LOAD FOR THE ENTIRE WEEK.
181  C
182                      IF(IIDAY.EQ.0)GO TO 125
183                      IF(IIDAY.EQ.1)MONTH1=MONTH
```

```
184                       IF(IIDAY.EQ.1)DAY1=DAY
185 C
186 C  MONTH1 AND DAY1 ARE THE MONTH AND DAY THAT THE CURRENT WEEK STARTS
187 C  MONTH2 AND DAY2 ARE THE MONTH AND DAY THAT THE PRESENTLY CALLED MAX.
188 C  LOAD WEEK STARTS.
189 C
190                       PEAKWK1(IIDAY,IHOUR)=LOAD(IHOUR)
191                       WKSUM1=WKSUM1+LOAD(IHOUR)
192 C
193 C  PEAKWK1 HOLDS THE CURRENT WEEK'S LOAD DATA
194 C  WKSUM1 HOLDS THE CURRENT WEEK'S LOAD DATA SUMMATION
195 C  WKSUM2 HOLDS THE LOAD DATA SUMMATION FOR THE MAXIMUM WEEK
196 C         ENCOUNTERED SO FAR
197 C
198   125                 CONTINUE
199                   IF(IIDAY.NE.7)GOTO115
200                   IF(WKSUM1.LT.WKSUM2)GO TO 6121
201 C
202 C  CHECK TO SEE IF THE CURRENT WEEK'S LOAD IS GREATER THAN THE WEEK
203 C  CURRENTLY CALLED THE MAXIMUM WEEK.
204 C     IF NOT, CONTINUE READING LOAD DATA
205 C     IF YES, SAVE THE CURRENT WEEK AS THE MAXIMUM WEEK AND CONTINUE
206 C             READING LOAD DATA
207 C
208                   WKSUM2=WKSUM1
209                   DAY2=DAY1
210                   MONTH2=MONTH1
211                       DO 135 I=1,7
212                           DO 145 J=1,24
213                           PEAKWK2(I,J)=PEAKWK1(I,J)
214   145                     CONTINUE
215   135                 CONTINUE
216  6121             IIDAY=WKSUM1=0
217   115             CONTINUE
218   105       CONTINUE
219 C
220 C  READ IN THE LOAD DATA FOR THE LAST FEW DAYS OF THE YEAR
221 C  IDAY365 HOLDS THE VALUE OF THE DAY ON WHICH DEC. 31 FALLS
222 C
223             DO 205 I=1,IDAY365
224             READ(23,820)MONTH,DAY,(LOAD(L),L=1,12)
225             READ(23,820)MONTH,DAY,(LOAD(L),L=13,24)
226                 DO 215 IHOUR=1,24
227   215           AVPEAK(I,IHOUR)=AVPEAK(I,IHOUR)+LOAD(IHOUR)
228   205       CONTINUE
229 C
230 C  CALCULATE THE AVERAGE HOURLY LOAD VALUES FROM THE ARRAY THAT HELD
231 C  THE HOURLY LOAD SUMS.
232 C
233             DO 305 IDAY=1,7
234             WEEKS=52.
235             IF(IDAY.EQ.IDAYONE.OR.IDAY.EQ.IDAY365)WEEKS=53.
236                 DO 315 IHOUR=1,24
237   315           AVPEAK(IDAY,IHOUR)=AVPEAK(IDAY,IHOUR)/WEEKS
238   305       CONTINUE
239         WRITE(2,920)MONTH2,DAY2
240 C
241 C  THE ARRAYS THAT HOLD THE VALUES FOR THE AVERAGE LOAD WEEK AND
242 C  THE PEAK LOAD WEEK ARE WRITTEN INTO FILES 4 AND 3 RESPECTIVELY
243 C
244             DO 405 IDAY=1,7
```

```
245           X="AM"
246           WRITE(3,923)IDAY,X,(PEAKWK2(IDAY,IHOUR),IHOUR=1,12)
247           X="PM"
248   405     WRITE(3,923)IDAY,X,(PEAKWK2(IDAY,IHOUR),IHOUR=13,24)
249           DO 505 IDAY=1,7
250           X="AM"
251           WRITE(4,923)IDAY,X,(AVPEAK(IDAY,IHOUR),IHOUR=1,12)
252           X="PM"
253   505     WRITE(4,923)IDAY,X,(AVPEAK(IDAY,IHOUR),IHOUR=13,24)
254       RETURN
255   801 FORMAT(A4)
256   820 FORMAT(I2,I2,16X,12F5.0)
257   920 FORMAT("-","THE MAXIMUM LOAD WEEK OCCURS IN MONTH ",I2,
258      +" BEGINNING ON DAY ",I2)
259   923 FORMAT("DAY ",I1,1X,A2,12(1X,F5.0))
260       END
261
262
263 C#############################################################################C
264 C#############################################################################C
265 C                                                                             C
266 C                        SUBROUTINE LDDIST                                    C
267 C                                                                             C
268 C#############################################################################C
269 C#############################################################################C
270
271
272       SUBROUTINE LDDIST
273 C
274 C LDDIST:  CALCULATES THE INVERTED LOAD DISTRIBUTION CURVE FOR BOTH
275 C          THE AVERAGE LOAD WEEK AND THE MAXIMUM LOAD WEEK
276 C
277       REAL LDPEAK(200,2)
278       REWIND 3
279       REWIND 4
280       REWIND 10
281       REWIND 11
282           DO 105 J=1,2
283           IFILE=J+2
284           LFILE=J+9
285                 DO 595 IX=1,14
286   595           READ(IFILE,830)(LDPEAK((IX-1)*12+IK,1),IK=1,12)
287 C
288 C READ IN THE HOURLY LOAD PEAK DATA FOR THE MAXIMUM WEEK FROM TAPE 3
289 C AND THE HOURLY LOAD PEAK DATA FOR THE AVERAGE WEEK FROM TAPE 4
290 C
291           LASTPAS=167
292 C
293 C THE LOOP ENDING AT STATEMENT 705 ORDERS THE HOURLY LOAD PEAK DATA
294 C FROM MAXIMUM LOAD TO MINIMUM LOAD
295 C
296                 DO 715 IHOUR=1,LASTPAS
297                 JHOUR=IHOUR+1
298                       DO 725 KHOUR=JHOUR,168
299                       IF(LDPEAK(IHOUR,1).GE.LDPEAK(KHOUR,1))
300      +                GO TO 725
301                       TEMPORY=LDPEAK(IHOUR,1)
302                       LDPEAK(IHOUR,1)=LDPEAK(KHOUR,1)
303                       LDPEAK(KHOUR,1)=TEMPORY
304   725                 CONTINUE
305   715           CONTINUE
```

```
306 C
307 C THE LOOP ENDING AT STATEMENT 1015 CALCULATES FOR BOTH LOAD
308 C DISTRIBUTIONS THE FRACTION OF THE WEEK THAT THE LOAD IS GREATER THAN
309 C OR EQUAL TO EACH DISCRETE MEGAWATT VALUE
310 C
311             REWIND LFILE
312             WRITE(LFILE,932)J,LDPEAK(1,1)+1.
313                  DO 1015 IHOURS=1,168
314                  IF(IHOURS.EQ.21.OR.IHOURS.EQ.42.OR.IHOURS.EQ.84.OR.
315       +          IHOURS.EQ.126.OR.IHOURS.EQ.168)WRITE(LFILE,932)J,LDPEAK
316       +          (IHOURS,1)
317  1015             CONTINUE
318   105       CONTINUE
319 C
320 C THE INVERTED LOAD DISTRIBUTION CURVE IS APPROXIMATED PIECEWISE
321 C LINEAR WITH SIX LINE SEGMENTS.  THE ENDPOINTS ARE CHOSEN WHERE THE
322 C THE FRACTION OF THE WEEK IS EQUAL TO 0 .125 .25 .5 .75 1.0 AND WHEN
323 C LOAD EQUALS 0 MW.
324 C
325         RETURN
326   830 FORMAT(8X,12(1X,F5.0))
327   932 FORMAT(1X,I2,2X,F10.2)
328         END
329
```

```
C#############################################################################C
C#############################################################################C
C                                                                             C
C          PROGRAM XPAN                          BY ERIC KRULL                C
C                                                                             C
C#############################################################################C
C#############################################################################C

      PROGRAM XPAN(INPUT,OUTPUT,DATIN=500B,DATOUT=500B,MAXPTS=500B,
C
C  PROGRAM XPAN USES THE FILES FORMED BY PREXPAN TO PERFORM THE
C  GENERATION EXPANSION PLANNING STUDY.
C
     +AVEPTS=500B,XFUEL=500B,XFCST=500B,XUBAS=500B,XUCAP=500B,XUFOR=500B
     +,XUHRT=500B,GLDOR=500B,BLDOR=500B,XLDOR=500B,XOPTN=500B,XDATA=500B
     +,TAPE1=DATIN,TAPE2=DATOUT,TAPE10=MAXPTS,TAPE11=AVEPTS,TAPE21=
     +XFUEL,TAPE22=XFCST,TAPE24=XUBAS,TAPE25=XUCAP,TAPE26=XUFOR,TAPE27=
     +XUHRT,TAPE30=GLDOR,TAPE31=BLDOR,TAPE32=XLDOR,TAPE40=XOPTN,TAPE41=
     +XDATA,TAPE6=OUTPUT,TAPE7=INPUT)
      DIMENSION I(4),ENTRY(30,30,4),ALSOMAX(20,4),INDEX(100),
     +NAME(100),NUMBER(100),UNIT(100,3),EQCURVE(200,2),IBESTOP(5,30),
     +HTRATE(7),CAPCTY(7),IFUEL(100),MLTSTAT(100),BENEFIT(30,5,2),
     +ECO(30)
      REAL LDORDER(100
     +,5),LDCURVE(7),MAXRET(5,30),ECTRY(30,30,4),ICAPSUM,IINDEX
      INTEGER YEAR,CAPTLMX,OPTION(4,6),XOPTION(30),STATE(30),BCOUNT,
     +ECOUNT,CONVLVD(100),XUNIT(100,3),NFLAG(100),CSTATE(100)
      COMMON I,ENTRY,CAPCTY,UNIT,EQCURVE,IBESTOP,ALSOMAX,BENEFIT,
     +LDORDER,MAXRET,XOPTION,CSTATE,IFUEL,MLTSTAT,ECO,ECTRY,
     +NFLAG,XUNIT,OPTION
      EQUIVALENCE (XUNIT(1,1),NAME),(XUNIT(1,2),NUMBER),(XUNIT(1,3),
     +INDEX),(LDCURVE,HTRATE,CAPCTY),(CONVLVD,NFLAG),(STATE,XOPTION)
      REWIND 1
C
C  INITIALIZE ALL STATES TO 0
C
            DO 1 NN=1,30
    1       STATE(NN)=0
        KCOUNT=MLTFLAG=0
        READ(1,800)IYEAR,GRWTHLD,CAPTLMX,IDAYONE,IDAY365,ESCRATE,IINDEX,
       +CAPRATE,XMWMAX,CLOLP,WK1,WK2
C
C  READ IN THE USER SUPPLIED DATA
C
        CALL OPTIONS(KOUNT,CAPTLMX)
C
C  FIND THE POSSIBLE EXPANSION OPTIONS THAT MEET THE CAPITAL MAXIMUM
C  CONSTRAINT.  KOUNT IS THE NUMBER OF EXPANSION OPTIONS.
C
C
C  GET THE NEEDED GENERATOR INFORMATION, AND SET UP FILES FOR THE BASE
C  SYSTEM AND THE EXPANSION UNITS.
C
            DO 105 YEAR=1,5
            IYEARR=IYEAR+YEAR
            CALL GENINFO(BCOUNT,ECOUNT,ICAPSUM,IYEARR)
            CALL GENRTRN(YEAR,KOUNT,BCOUNT,ECOUNT,ESCRATE,CAPTLMX,
     +      ICAPSUM,GRWTHLD,IINDEX,CAPRATE,XMWMAX,CLOLP,WK1,WK2)
C
C  FOR THE FIVE YEARS OF THE STUDY, CALCULATE THE RETURNS FOR EVERY
```

```
C  POSSIBLE STATE OF THE SYSTEM USING THE PROPER UTILITY FUNCTIONS
C
  105     CONTINUE
        CALL MAXSTAT(KOUNT,MLTFLAG,KCOUNT)
C
C  USING THE RETURNS CALCULATED IN GENRTRN, CALCULATE THE MULTI-
C  ATTRIBUTE UTILITY FOR EVERY STATE IN EACH OF THE FIVE YEARS.
C  THEN FIND THE MAXIMUM RETURNS FOR EACH EXPANSION STATE AND THE
C  BEGINNING STATE (OF THAT YEAR) THAT LED TO THESE MAXIMUM RETURNS.
C
  800 FORMAT(I2,1X,F6.4,1X,I7,2(1X,I1),1X,F6.4,1X,F6.1,1X,F6.4,1X,F5.0,
     +1X,F8.5,1X,F5.3,1X,F5.3)
        STOP
        END


C##############################################################################C
C##############################################################################C
C                                                                              C
C                          SUBROUTINE OPTIONS                                  C
C                                                                              C
C##############################################################################C
C##############################################################################C


        SUBROUTINE OPTIONS(KOUNT,CAPTLMX)
C
C  SUBROUTINE OPTIONS GENERATES ALL POSSIBLE EXPANSION OPTIONS (STATES)
C
        DIMENSION I(4),ENTRY(30,30,4),ALSOMAX(20,4),INDEX(100),
       +NAME(100),NUMBER(100),UNIT(100,3),EQCURVE(200,2),IBESTOP(5,30),
       +HTRATE(7),CAPCTY(7),IFUEL(100),MLTSTAT(100),ECO(30),BENEFIT(30,
       +5,2)
        REAL LDORDER(100
       +,5),LDCURVE(7),MAXRET(5,30),ECTRY(30,30,4),ICAPSUM
        INTEGER YEAR,CAPTLMX,OPTION(4,6),XOPTION(30),STATE(30),BCOUNT,
       +ECOUNT,COUNT,CONVLVD(100),XUNIT(100,3),NFLAG(100),CSTATE(100)
        COMMON I,ENTRY,CAPCTY,UNIT,EQCURVE,IBESTOP,ALSOMAX,BENEFIT,
       +LDORDER,MAXRET,XOPTION,CSTATE,IFUEL,MLTSTAT,ECO,ECTRY,
       +NFLAG,XUNIT,OPTION
        EQUIVALENCE (XUNIT(1,1),NAME),(XUNIT(1,2),NUMBER),(XUNIT(1,3),
       +INDEX),(LDCURVE,HTRATE,CAPCTY),(CONVLVD,NFLAG),(STATE,XOPTION)
        REWIND 40
        REWIND 41
C
C  READ IN ALL OF THE GENERATOR EXPANSION UNIT DATA
C
            DO 105 IK=1,4
  105       READ (41,900)(OPTION(IK,IL),IL=1,6)
        LOOP1=OPTION(1,2)+1
        LOOP2=OPTION(2,2)+1
        LOOP3=OPTION(3,2)+1
        LOOP4=OPTION(4,2)+1
C
C  LOOPS 215,225,235,245 GENERATE ALL POSSIBLE COMBINATIONS OF EX-
C  PANSION UNITS THAT DON'T VIOLATE THE MAXIMUM NUMBER OF EACH UNIT
C  CONSTRAINT
C
            DO 215 I1=1,LOOP1
                 DO 225 I2=1,LOOP2
                      DO 235 I3=1,LOOP3
```

```
123                           DO 245 I4=1,LOOP4
124                           I11=I1-1
125                           I21=I2-1
126                           I31=I3-1
127                           I41=I4-1
128                           IF(I11*OPTION(1,1)+I21*OPTION(2,1)+I31*OPTION(
129         +                 3,1)+I41*OPTION(4,1).LE.CAPTLMX)WRITE(40,940)
130         +                 I11,I21,I31,I41
131 C
132 C  CHECK TO SEE IF THIS COMBINATION OF EXPANSION UNITS VIOLATES THE
133 C  MAXIMUM CAPITAL CONSTRAINT.
134 C
135    245                        CONTINUE
136    235                  CONTINUE
137    225             CONTINUE
138    215        CONTINUE
139           WRITE(40,940)9,9,9,9
140 C
141 C  LOOPING OVER, SO WRITE OUT A CODE THAT SIGNIFIES THE LAST EXPANSION
142 C  OPTION
143 C
144           KOUNT=0
145           REWIND 40
146 C
147 C  REWIND THE FILE THAT HOLDS THE EXPANSION OPTIONS SO THEY CAN BE
148 C  COUNTED
149 C
150        61 READ(40,940)I1,I2,I3,I4
151           BACKSPACE 40
152           READ(40,941)ISTATE
153 C
154 C  I1,I2,I3,I4 ARE THE NUMBER OF EACH UNIT INSTALLED FOR THIS OPTION.
155 C  ISTATE IS THE 4-DIGIT STATE REPRESENTATION
156 C
157           IF(ISTATE.EQ.9999)GOTO62
158           KOUNT=KOUNT+1
159           IF(KOUNT.GT.30)GOTO63
160           XOPTION(KOUNT)=ISTATE
161 C
162 C  SAVE THE EXPANSION OPTIONS IN THE XOPTION ARRAY
163 C
164           GOTO61
165        62 RETURN
166        63 WRITE(2,960)
167           STOP 03
168       900 FORMAT(I7,1X,I2,1X,I2,1X,I4,1X,A8,1X,I2)
169       940 FORMAT(1X,4I1)
170       941 FORMAT(1X,I4)
171       960 FORMAT("ERROR:  THE POSSIBLE XPAN STATES EXCEED 30 PER YEAR")
172           END
173
174
175 C#########################################################################C
176 C#########################################################################C
177 C                                                                         C
178 C                        SUBROUTINE GENINFO                               C
179 C                                                                         C
180 C#########################################################################C
181 C#########################################################################C
182
183
```

```
      SUBROUTINE GENINFO(BCOUNT,ECOUNT,ICAPSUM,IYEAR)
C
C  SUBROUTINE GENINFO TAKES THE GENERATOR INFORMATION FILES CREATED
C  BY PREXPAN, AND FINDS THE INFORMATION IT NEEDS TO CALCULATE
C  THE INCREMENTAL FUEL COST FOR EACH CAPACITY STATE.
C  GENINFO ALSO CREATES SEPERATE FILES FOR THE BASE UNIT'S AND EXPANSION
C  UNIT'S CAPACITY STATES
C
      DIMENSION I(4),ENTRY(30,30,4),ALSOMAX(20,4),INDEX(100),
     +NAME(100),NUMBER(100),UNIT(100,3),EQCURVE(200,2),IBESTOP(5,30),
     +HTRATE(7),CAPCTY(7),IFUEL(100),MLTSTAT(100),ECO(30),BENEFIT(30,
     +5,2)
      REAL LDORDER(100
     +,5),LDCURVE(7),MAXRET(5,30),ECTRY(30,30,4),ICAPSUM
      INTEGER YEAR,CAPTLMX,OPTION(4,6),XOPTION(30),STATE(30),BCOUNT,
     +ECOUNT,COUNT,CONVLVD(100),XUNIT(100,3),NFLAG(100),CSTATE(100)
      COMMON I,ENTRY,CAPCTY,UNIT,EQCURVE,IBESTOP,ALSOMAX,BENEFIT,
     +LDORDER,MAXRET,XOPTION,CSTATE,IFUEL,MLTSTAT,ECO,ECTRY,
     +NFLAG,XUNIT,OPTION
      EQUIVALENCE (XUNIT(1,1),NAME),(XUNIT(1,2),NUMBER),(XUNIT(1,3),
     +INDEX),(LDCURVE,HTRATE,CAPCTY),(CONVLVD,NFLAG),(STATE,XOPTION)
      REWIND 22
      REWIND 24
      REWIND 25
      REWIND 26
      REWIND 27
      REWIND 30
      REWIND 31
      REWIND 32
      LAST="LAST"
      Q="Q"
      BCOUNT=0
C
C  BCOUNT HOLDS THE NUMBER OF CAPACITY STATES FOR THE BASE SYSTEM
C  ECOUNT HOLDS THE NUMBER OF CAPACITY STATES FOR THE EXPANSION UNITS
C
      COUNT=0
      ICAPSUM=0
 6001 READ(25,905)NNAME,NNUMBER,IXFLAG,(CAPCTY(L),L=1,7)
C
C  READ IN THE UNIT CAPACITY STATES FOR THIS GENERATOR AND THE FLAG
C  THAT SPECIFIES WHETHER IT IS A BASE OR EXPANSION UNIT
C
      IF(NNAME.EQ.LAST)GOTO6002
      KOUNT=0
          DO 115 ISTATE=1,7
          IF(CAPCTY(ISTATE).EQ.0.)GOTO6111
  115     KOUNT=KOUNT+1
 6111     DO 205 ICOUNT=1,KOUNT
          IZ=COUNT+ICOUNT
          NAME(IZ)=NNAME
          NUMBER(IZ)=NNUMBER
          NFLAG(IZ)=IXFLAG
          LDORDER(IZ,1)=CAPCTY(ICOUNT)
          IF(ICOUNT.GT.1)MLTSTAT(IZ)=0
          IF(ICOUNT.EQ.1)MLTSTAT(IZ)=1
          IF(ICOUNT.EQ.KOUNT.AND.IXFLAG.NE.1)ICAPSUM=ICAPSUM+CAPCTY(
     +    ICOUNT)
  205     CSTATE(IZ)=ICOUNT
      COUNT=COUNT+KOUNT
      IF(IXFLAG.NE.1)BCOUNT=BCOUNT+KOUNT
```

```
      ECOUNT=COUNT-BCOUNT
      IF(COUNT.GT.100)WRITE(6,990)
      GOTO6001
C
C  BEGINNING WITH THE NEXT STATEMENT, THE REST OF THE STATEMENTS IN THIS
C  SUBROUTINE CALCULATE THE INCREMENTAL FUEL COST FOR EACH CAPACITY
C  STATE USING THE INFORMATION FILES CREATED BY PREXPAN
C
 6002 READ(27,907)NNAME,NNUMBER,XX,(HTRATE(L),L=1,7)
      IF(NNAME.EQ.LAST)GOTO6003
          DO 305 IFIND=1,COUNT
          IF(NAME(IFIND).EQ.NNAME.AND.NUMBER(IFIND).EQ.NNUMBER)
     +    GOTO6301
  305     CONTINUE
 6301 IF(XX.EQ.Q)LDORDER(IFIND,2)=(HTRATE(3)*LDORDER(IFIND,1)+HTRATE(2))
     +*LDORDER(IFIND,1)+HTRATE(1)
      IF(XX.NE.Q)LDORDER(IFIND,2)=HTRATE(CSTATE(IFIND))/1000.*LDORDER(
     +IFIND,1)
      IFIND=IFIND+1
      IF(NNAME.EQ.NAME(IFIND).AND.NNUMBER.EQ.NUMBER(IFIND))GOTO6301
      GOTO6002
 6003 READ(26,906)NNAME,NNUMBER,FORATE
      IF(NNAME.EQ.LAST)GOTO6004
          DO 405 IFIND=1,COUNT
          IF(NAME(IFIND).EQ.NNAME.AND.NUMBER(IFIND).EQ.NNUMBER)
     +    GOTO6401
  405     CONTINUE
 6401 LDORDER(IFIND,3)=FORATE
      IFIND=IFIND+1
      IF(NAME(IFIND).EQ.NNAME.AND.NUMBER(IFIND).EQ.NNUMBER)GOTO6401
      GOTO6003
 6004 READ(24,904)NNAME,NNUMBER,LFUEL,FXOPCOS,VARCOST
      IF(NNAME.EQ.LAST)GOTO6005
          DO 505 IFIND=1,COUNT
          IF(NAME(IFIND).EQ.NNAME.AND.NUMBER(IFIND).EQ.NNUMBER)GOTO6501
  505     CONTINUE
 6501 IFUEL(IFIND)=LFUEL
      LDORDER(IFIND,4)=FXOPCOS
      LDORDER(IFIND,5)=VARCOST
      IFIND=IFIND+1
      IF(NAME(IFIND).EQ.NNAME.AND.NUMBER(IFIND).EQ.NNUMBER)GOTO6501
      GOTO6004
 6005     DO 775 INUM=1,COUNT
 6771     READ(22,902)IFFUEL,IYYEAR,FCOST
          IF(IFUEL(INUM).NE.IFFUEL.OR.IYEAR.NE.IYYEAR)GOTO6771
C
C  LDORDER(INUM,2) HOLDS THE INCREMENTAL FUEL COST FOR CAPACITY STATE
C  (INUM).  THIS INFORMATION ALONG WITH THE UNIT NAME,NUMBER,FUEL,
C  CAPACITY, FIXED COSTS, OPERATING COSTS, ETC... ARE PRINTED INTO
C  EITHER THE BASE OR EXPANSION UNIT FILES FOR LATER USE.
C
          LDORDER(INUM,2)=(FCOST*LDORDER(INUM,2)/100.+LDORDER(INUM,4)
     +    *1000./168.)/LDORDER(INUM,1)+LDORDER(INUM,5)
  775     REWIND 22
          DO 785 III=1,COUNT
          IF(NFLAG(III).EQ.1)KFILE=32
          IF(NFLAG(III).NE.1)KFILE=31
  785     WRITE(KFILE,930)NAME(III),NUMBER(III),CSTATE(III),MLTSTAT(
     +    III),IFUEL(III),(LDORDER(III,IK),IK=1,5)
      RETURN
  902 FORMAT(2(I2,2X),3X,F7.1)
```

```
  904 FORMAT(A8,I2,2X,I3,2X,F5.2,2X,F5.3)
  905 FORMAT(A8,I2,1X,I1,F6.0,6(2X,F6.0))
  906 FORMAT(A8,I2,2X,F5.1)
  907 FORMAT(A8,I2,2X,A1,2X,F6.0,2(2X,F7.0),4(2X,F6.0))
  930 FORMAT(A8,4(I2,2X),F5.0,2X,F10.5,2X,F7.4,2X,F5.2,2X,F5.3)
  990 FORMAT("-","ERROR:    NUMBER OF GENERATION CAPACITY STATES EXCEEDS"
     +," 100")
      END


C#######################################################################C
C#######################################################################C
C                                                                       C
C                          SUBROUTINE GENRTRN                           C
C                                                                       C
C#######################################################################C
C#######################################################################C


      SUBROUTINE GENRTRN(YEAR,NOPTS,BCOUNT,ECOUNT,ESCRATE,CAPTLMX,
C
C  THIS SUBROUTINE CALCULATES THE RETURNS FOR THE THREE COMPONENT
C  UTILITY FUNCTIONS FOR EVERY STATE IN THE GIVEN YEAR
C
     +ICAPSUM,GRWTHLD,IINDEX,CAPRATE,XMWMAX,CLOLP,WK1,WK2)
      DIMENSION I(4),ENTRY(30,30,4),ALSOMAX(20,4),INDEX(100),
     +NAME(100),NUMBER(100),UNIT(100,3),EQCURVE(200,2),IBESTOP(5,30),
     +HTRATE(7),CAPCTY(7),IFUEL(100),MLTSTAT(100),BENEFIT(30,5,2),
     +ECO(30)
      REAL LDORDER(100
     +,5),LDCURVE(7),MAXRET(5,30),ECTRY(30,30,4),ICAPSUM,MAXLOLP,IINDEX
      INTEGER YEAR,CAPTLMX,OPTION(4,6),XOPTION(30),STATE(30),BCOUNT,
     +ECOUNT,COUNT,CONVLVD(100),XUNIT(100,3),NFLAG(100),CSTATE(100)
      COMMON I,ENTRY,CAPCTY,UNIT,EQCURVE,IBESTOP,ALSOMAX,BENEFIT,
     +LDORDER,MAXRET,XOPTION,CSTATE,IFUEL,MLTSTAT,ECO,ECTRY,
     +NFLAG,XUNIT,OPTION
      EQUIVALENCE (XUNIT(1,1),NAME),(XUNIT(1,2),NUMBER),(XUNIT(1,3),
     +INDEX),(LDCURVE,HTRATE,CAPCTY),(CONVLVD,NFLAG),(STATE,XOPTION)
           DO 115 IOPTION=1,NOPTS
           REWIND 40
           WRITE(40,941)XOPTION(IOPTION)
           BACKSPACE 40
           READ(40,940)(I(K),K=1,4)
C
C  READ IN AN EXPANSION OPTION AND STORE THE NUMBER OF EACH EXPANSION
C  UNIT TO INSTALL IN ARRAY I.
C
           REWIND 30
           REWIND 31
                DO 125 IGEN=1,BCOUNT
                READ(31,930)NNAME,NNUMBER,KSTATE,MLT,MFUEL,D1,D2,
     +          D3,D4,D5
  125           WRITE(30,930)NNAME,NNUMBER,KSTATE,MLT,MFUEL,D1,D2,
     +          D3,D4,D5
C
C  READ IN THE BASE SYSTEM GENERATOR DATA, AND WRITE IT INTO FILE GLDOR
C
           LCOUNT=0
                DO 135 KK=1,4
                IF(I(KK).EQ.0)GOTO135
                IKK=I(KK)
```

```
                    DO 145 L=1,IKK
                    REWIND 32
                         DO 155 ICOUNT=1,ECOUNT
                         READ(32,930)NNAME,NNUMBER,KSTATE,MLT,MFUEL
     +                   ,D1,D2,D3,D4,D5
                         IF(NNAME.NE.OPTION(KK,5).OR.NNUMBER.NE.
     +                   OPTION(KK,6))GOTO155
                         WRITE(30,930)NNAME,NNUMBER+L-1,KSTATE,MLT,
     +                   MFUEL,D1,D2,D3,D4,D5
C
C  WRITE THE PROPER NUMBER OF EACH EXPANSION UNIT AND THE CORRESPON-
C  DING GENERATOR INFORMATION INTO THE GLDOR FILE.
C
                         LCOUNT=LCOUNT+1
  155                    CONTINUE
  145               CONTINUE
  135          CONTINUE
C
C  CALCULATE THE GENERATION CAPACITY FOR THIS EXPANSION STATE
C
            GCAPSUM=ICAPSUM+I(1)*OPTION(1,4)+I(2)*OPTION(2,4)+I(3)*
     +      OPTION(3,4)+I(4)*OPTION(4,4)
            COUNT=LCOUNT+BCOUNT
            CALL ENVRNMT(ENVBNFT,XMWMAX)
C
C  CALCULATE THE ENVIRONMENTAL BENEFIT FOR THIS EXPANSION STATE
C
            CALL ORDER(COUNT)
C
C  PUT THE GENERAOR CAPACITY STATES IN DISPATCH ORDER
C
            CALL LOLP(GCAPSUM,MAXLOLP,AVELOLP,LASTPT,YEAR,GRWTHLD,
     +      COUNT,IINDEX)
C
C  CALCULATE THE LOLP FOR THE MAXIMUM  AND  AVERAGE LOAD WEEKS
C
            CALL RELBLTY(MAXLOLP,AVELOLP,RELBNFT,CLOLP,WK1,WK2)
C
C  CALCULATE THE RELIBILITY BENEFIT, AND STORE IN ARRAY BENEFIT(,,2)
C
            BENEFIT(IOPTION,YEAR,1)=ENVBNFT
C
C  STORE THE ENVIRONMENTAL BENEFIT IN ARRAY BENEFIT(,,1)
C
            BENEFIT(IOPTION,YEAR,2)=RELBNFT
                 DO 165 LSTATE=1,NOPTS
C
C  CALCULATE THE ECONOMIC BENEFIT IN GOING FROM ANY PREVIOUS SYSTEM
C  STATE TO THIS POSSIBLE EXPANSION STATE.
C
                 WRITE(40,941)STATE(LSTATE)
                 BACKSPACE 40
                 READ(40,940)L1,L2,L3,L4
                 IF(I(1).LT.L1.OR.I(2).LT.L2.OR.I(3).LT.L3.OR.I(4).LT.L4)
     +           GOTO6161
                 CALL ECONOMC(YEAR,CAPTLMX,ECOBNFT,ESCRATE,L1,L2,L3,L4,
     +           CAPRATE)
C
C  STORE THE ECONOMIC BENEFITS FOR YEAR 1 IN ARRAY ECO, AND FOR YEARS
C  2-5 IN ARRAY ECTRY.
C
```

```
428                IF(YEAR.EQ.1)ECO(IOPTION)=ECOBNFT
429                IF(YEAR.GT.1)ECTRY(LSTATE,IOPTION,YEAR-1)=ECOBNFT
430                IF(YEAR.EQ.1)GOTO115
431                GOTO165
432  6161          ECTRY(LSTATE,IOPTION,YEAR-1)=0.0
433 C
434 C  IF AN EXPANSION OPTION VIOLATES A CONSTRAINT, STORE THE ECONOMIC
435 C  BENEFIT AS 0.0
436 C
437   165          CONTINUE
438   115      CONTINUE
439        RETURN
440   930 FORMAT(A8,4(I2,2X),F5.0,2X,F10.5,2X,F7.4,2X,F5.2,2X,F5.3)
441   940 FORMAT(1X,411)
442   941 FORMAT(1X,I4)
443        END
444
445
446 C#########################################################################C
447 C#########################################################################C
448 C                                                                         C
449 C                        SUBROUTINE ECONOMC                               C
450 C                                                                         C
451 C#########################################################################C
452 C#########################################################################C
453
454
455        SUBROUTINE ECONOMC(YEAR,CAPTLMX,ECOBNFT,ESCRATE,L1,L2,L3,L4,
456       +CAPRATE)
457 C
458 C  THIS SUBROUTINE CALCULATES THE ECONOMIC BENEFIT AS SPECIFIED BY THE
459 C  ECONOMIC COMPONENT UTILITY FUNCTION
460 C
461        DIMENSION I(4),ENTRY(30,30,4),ALSOMAX(20,4),INDEX(100),
462       +NAME(100),NUMBER(100),UNIT(100,3),EQCURVE(200,2),IBESTOP(5,30),
463       +HTRATE(7),CAPCTY(7),IFUEL(100),MLTSTAT(100),ECO(30),BENEFIT(30,
464       +5,2)
465        REAL LDORDER(100
466       +,5),LDCURVE(7),MAXRET(5,30),ECTRY(30,30,4),ICAPSUM
467        INTEGER YEAR,CAPTLMX,OPTION(4,6),XOPTION(30),STATE(30),BCOUNT,
468       +ECOUNT,COUNT,CONVLVD(100),XUNIT(100,3),NFLAG(100),CSTATE(100)
469        COMMON I,ENTRY,CAPCTY,UNIT,EQCURVE,IBESTOP,ALSOMAX,BENEFIT,
470       +LDORDER,MAXRET,XOPTION,CSTATE,IFUEL,MLTSTAT,ECO,ECTRY,
471       +NFLAG,XUNIT,OPTION
472        EQUIVALENCE (XUNIT(1,1),NAME),(XUNIT(1,2),NUMBER),(XUNIT(1,3),
473       +INDEX),(LDCURVE,HTRATE,CAPCTY),(CONVLVD,NFLAG),(STATE,XOPTION)
474        ESC=(1.0+ESCRATE)**YEAR
475        CAP=(1.0+CAPRATE)**YEAR
476 C
477 C  CAP IS THE ESCALATION FACTOR FOR THE CAPITAL AVAILABLE
478 C  ESC IS THE ESCALATION FACTOR FOR THE CONSTRUCTION COSTS
479 C
480        OPTCOST=((I(1)-L1)*OPTION(1,1)+(I(2)-L2)*OPTION(2,1)+(I(3)-L3)*
481       +OPTION(3,1)+(I(4)-L4)*OPTION(4,1))*ESC
482        CAPITAL=CAPTLMX*CAP
483        IF(CAPITAL.LT.OPTCOST)GOTO6
484 C
485 C  OPTCOST IS THE COST INCURRED IN GOING FROM THE BEGINNING STATE
486 C  TO THE EXPANSION STATE FOR THIS STAGE.
487 C
488        ECOBNFT=1.0-OPTCOST/CAPITAL
```

```
C
C  ECOBNFT IS THE ECONOMIC BENEFIT CALCULATED BY THE UTILITY FUNCTION
C
      RETURN
    6 ECOBNFT=0.0
      RETURN
      END


C##############################################################################C
C##############################################################################C
C                                                                              C
C                        SUBROUTINE ENVRNMT                                    C
C                                                                              C
C##############################################################################C
C##############################################################################C


      SUBROUTINE ENVRNMT(ENVBNFT,XMWMAX)
C
C  THIS SUBROUTINE CALCULATES THE ENVIRONMENTAL BENEFIT AS SPECIFIED
C  BY THE ENVIRONMENTAL COMPONENT UTILITY FUNCTION
C
      DIMENSION I(4),ENTRY(30,30,4),ALSOMAX(20,4),INDEX(100),
     +NAME(100),NUMBER(100),UNIT(100,3),EQCURVE(200,2),IBESTOP(5,30),
     +HTRATE(7),CAPCTY(7),IFUEL(100),MLTSTAT(100),ECO(30),BENEFIT(30,
     +5,2)
      REAL LDORDER(100
     +,5),LDCURVE(7),MAXRET(5,30),ECTRY(30,30,4),ICAPSUM
      INTEGER YEAR,CAPTLMX,OPTION(4,6),XOPTION(30),STATE(30),BCOUNT,
     +ECOUNT,COUNT,CONVLVD(100),XUNIT(100,3),NFLAG(100),CSTATE(100)
      COMMON I,ENTRY,CAPCTY,UNIT,EQCURVE,IBESTOP,ALSOMAX,BENEFIT,
     +LDORDER,MAXRET,XOPTION,CSTATE,IFUEL,MLTSTAT,ECO,ECTRY,
     +NFLAG,XUNIT,OPTION
      EQUIVALENCE (XUNIT(1,1),NAME),(XUNIT(1,2),NUMBER),(XUNIT(1,3),
     +INDEX),(LDCURVE,HTRATE,CAPCTY),(CONVLVD,NFLAG),(STATE,XOPTION)
      SUM=0.
          DO 205 N=1,4
  205     SUM=SUM+OPTION(N,3)*OPTION(N,4)*I(N)/10.
      IF(SUM.GT.XMWMAX)GOTO6
      ENVBNFT=1.0-SUM/XMWMAX
C
C  THE LAST TWO LINES CALCULATE THE UTILITY; USING THE USER SPECIFIED
C  VALUE FOR XMWMAX. OPTION(,3)=ENV IMPACT NO.  OPTION(,4)=CAPCITY
C  I(N)= NO. OF UNITS N TO BE INSTALLED
C
      RETURN
    6 ENVBNFT=0.0
      RETURN
      END


C##############################################################################C
C##############################################################################C
C                                                                              C
C                        SUBROUTINE ORDER                                      C
C                                                                              C
C##############################################################################C
C##############################################################################C


```

```
      SUBROUTINE ORDER(COUNT)
C
C  THIS SUBROUTINE ORDERS THE GENERATOR CAPACITY STATES BY INCRE-
C  MENTAL FUEL COSTS
C
      DIMENSION I(4),ENTRY(30,30,4),ALSOMAX(20,4),INDEX(100),
     +NAME(100),NUMBER(100),UNIT(100,3),EQCURVE(200,2),IBESTOP(5,30),
     +HTRATE(7),CAPCTY(7),IFUEL(100),MLTSTAT(100),ECO(30),BENEFIT(30,
     +5,2)
      REAL LDORDER(100
     +,5),LDCURVE(7),MAXRET(5,30),ECTRY(30,30,4),ICAPSUM
      INTEGER YEAR,CAPTLMX,OPTION(4,6),XOPTION(30),STATE(30),BCOUNT,
     +ECOUNT,COUNT,CONVLVD(100),XUNIT(100,3),NFLAG(100),CSTATE(100),
     +CCOUNT
      COMMON I,ENTRY,CAPCTY,UNIT,EQCURVE,IBESTOP,ALSOMAX,BENEFIT,
     +LDORDER,MAXRET,XOPTION,CSTATE,IFUEL,MLTSTAT,ECO,ECTRY,
     +NFLAG,XUNIT,OPTION
      EQUIVALENCE (XUNIT(1,1),NAME),(XUNIT(1,2),NUMBER),(XUNIT(1,3),
     +INDEX),(LDCURVE,HTRATE,CAPCTY),(CONVLVD,NFLAG),(STATE,XOPTION)
      REWIND 30
          DO 805 IA=1,COUNT
          READ(30,930)NAME(IA),NUMBER(IA),CSTATE(IA),MLTSTAT(IA),
     +    IFUEL(IA),(LDORDER(IA,IM),IM=1,5)
C
C  READ THE INFORMATION ASSOCIATED WITH EACH CAPACITY STATE
C
  805     INDEX(IA)=IA
      CCOUNT=COUNT-1
C
C  USING AN INDEX SORT, ORDER THE CAPACITY STATES BY INCREMENTAL FUEL
C  COST (LDORDER(,2)), WITH THE ADDITIONAL CONSTRAINT THAT THE SMALLEST
C  CAPACITY STATE OF A UNIT BE DISPATCHED FIRST,FOLLOWED BY THE NEXT
C  LARGEST CAPACITY STATE, ETC...
C
          DO 605 INUM=1,CCOUNT
          JCOUNT=INUM+1
          IF(MLTSTAT(INDEX(INUM)).EQ.1)GOTO6601
          IN=INUM+1
               DO 665 IIN=IN,COUNT
               IF(MLTSTAT(INDEX(IIN)).EQ.0)GOTO665
               TEMP=INDEX(IIN)
               INDEX(IIN)=INDEX(INUM)
               INDEX(INUM)=TEMP
               GOTO6601
  665          CONTINUE
 6601          DO 615 KCOUNT=JCOUNT,COUNT
               IF(MLTSTAT(INDEX(KCOUNT)).EQ.0)GOTO615
               IF(LDORDER(INDEX(INUM),2).LE.LDORDER(INDEX(KCOUNT),2))
     +         GOTO615
               TEMPORY=INDEX(INUM)
               INDEX(INUM)=INDEX(KCOUNT)
               INDEX(KCOUNT)=TEMPORY
  615          CONTINUE
          MLTSTAT(INDEX(INUM))=0
          ICHANGE=INDEX(INUM)+1
          IF(ICHANGE.GT.COUNT)GOTO605
          IF(NAME(ICHANGE).EQ.NAME(INDEX(INUM)).AND.NUMBER(INDEX
     +    (INUM)).EQ.NUMBER(ICHANGE).AND.CSTATE(ICHANGE).EQ.
     +    CSTATE(INDEX(INUM))+1)MLTSTAT(ICHANGE)=1
  605     CONTINUE
      REWIND 30
```

```
611 C
612 C  WRITE OUT THE ORDERED CAPACITY STATES IN FILE GLDOR
613 C
614             DO 705 IORDER=1,COUNT
615   705       WRITE(30,930)NAME(INDEX(IORDER)),NUMBER(INDEX(IORDER)),
616      +      CSTATE(INDEX(IORDER)),MLTSTAT(INDEX(IORDER)),IFUEL(INDEX(
617      +      IORDER)),(LDORDER(INDEX(IORDER),IK),IK=1,5)
618         RETURN
619   930 FORMAT(A8,4(I2,2X),F5.0,2X,F10.5,2X,F7.4,2X,F5.2,2X,F5.3)
620         END
621
622
623 C#########################################################################
624 C#########################################################################
625 C
626 C                          SUBROUTINE LOLP
627 C
628 C#########################################################################
629 C#########################################################################
630
631
632         SUBROUTINE LOLP(ICAPSUM,MAXLOLP,AVELOLP,LASTPT,YEAR,GRWTHLD,
633        +NUNITS,XINDEX)
634 C
635 C SUBROUTINE LOLP CALCULATES THE LOSS OF LOAD PROBABILITY (LOLP) OF
636 C THE PRESENT GENERATION CONFIGURATION, FOR BOTH THE MAXIMUM AND AVERAGE
637 C LOAD WEEKS.
638 C
639         DIMENSION M(4),ENTRY(30,30,4),ALSOMAX(20,4),INDEX(100),
640        +NAME(100),NUMBER(100),UNIT(100,3),EQCURVE(200,2),IBESTOP(5,30),
641        +HTRATE(7),CAPCTY(7),IFUEL(100),MLTSTAT(100),ECO(30),BENEFIT(30,
642        +5,2)
643         REAL LDORDER(100,5),MAXLOLP,
644        +LDCURVE(7),MAXRET(5,30),ECTRY(30,30,4),ICAPSUM,MW,IINDEX
645         INTEGER YEAR,CAPTLMX,OPTION(4,6),XOPTION(30),STATE(30),BCOUNT,
646        +ECOUNT,COUNT,CONVLVD(100),XUNIT(100,3),NFLAG(100),CSTATE(100)
647         COMMON M,ENTRY,CAPCTY,UNIT,EQCURVE,IBESTOP,ALSOMAX,BENEFIT,
648        +LDORDER,MAXRET,XOPTION,CSTATE,IFUEL,MLTSTAT,ECO,ECTRY,
649        +NFLAG,XUNIT,OPTION
650         EQUIVALENCE (XUNIT(1,1),NAME),(XUNIT(1,2),NUMBER),(XUNIT(1,3),
651        +INDEX),(LDCURVE,HTRATE,CAPCTY),(CONVLVD,NFLAG),(STATE,XOPTION)
652         REWIND 10
653         REWIND 11
654           DO 95 J=1,2
655             DO 105 K=1,6
656             LFILE=J+9
657             READ(LFILE,832)JCURVE,LDCURVE((7-K))
658             LDCURVE((7-K))=LDCURVE((7-K))*((1.+GRWTHLD)**YEAR)
659   105       CONTINUE
660 C
661 C READ THE SIX POINTS SAVED FROM THE INVERTED LOAD DISTRIBUTION CURVE
662 C
663 C
664 C TO CALCULATE THE LOLP, THE EQUIVALENT LOAD DISTRIBUTION CURVE MUST
665 C BE CONSTRUCTED, TO TAKE INTO ACCOUNT THE PROBABILITY OF EACH GENERATOR
666 C BEING UNAVAILABLE.
667 C
668         REWIND 30
669         DO 5 ICOUNT=1,NUNITS
670         READ(30,930)NNAME,NNUMBER,NSTATE,CAP,XFOR
671             XUNIT(ICOUNT,1)=NNAME
```

```
672           XUNIT(ICOUNT,2)=NNUMBER
673           XUNIT(ICOUNT,3)=NSTATE
674         UNIT(ICOUNT,1)=CAP
675         UNIT(ICOUNT,2)=XFOR/100.
676       5 UNIT(ICOUNT,3)=1-UNIT(ICOUNT,2)
677 C
678 C THE EQUIVALENT CURVE IS APPROXIMATED BY TAKING A POINT FROM THE
679 C PREVIOUS CURVE EVERY "IINDEX" MW.
680 C
681         IINDEX=(LDCURVE(6)-LDCURVE(1))/4.
682         IF(XINDEX.LT.IINDEX)IINDEX=XINDEX
683         MW=LDCURVE(1)-IINDEX
684         LASTPT=INT((ICAPSUM+LDCURVE(6)-LDCURVE(1))/IINDEX)+1
685             DO 2005 NUMBERR=1,LASTPT
686             MW=MW+IINDEX
687             EQCURVE(NUMBERR,1)=MW
688             IF(MW.LE.LDCURVE(2))EQCURVE(NUMBERR,2)=(MW-LDCURVE(1))/
689       +     (LDCURVE(2)-LDCURVE(1))*(0.-.25)+1.
690             IF(MW.GT.LDCURVE(2).AND.MW.LE.LDCURVE(3))EQCURVE(NUMBERR,2)
691       +     =(MW-LDCURVE(2))/(LDCURVE(3)-LDCURVE(2))*(0-.25)+.75
692             IF(MW.GT.LDCURVE(3).AND.MW.LE.LDCURVE(4))EQCURVE(NUMBERR,2)
693       +     =(MW-LDCURVE(3))/(LDCURVE(4)-LDCURVE(3))*(0-.25)+.5
694             IF(MW.GT.LDCURVE(4).AND.MW.LE.LDCURVE(5))EQCURVE(NUMBERR,2)
695       +     =(MW-LDCURVE(4))/(LDCURVE(5)-LDCURVE(4))*(0-.125)+.25
696             IF(MW.GT.LDCURVE(5).AND.MW.LE.LDCURVE(6))EQCURVE(NUMBERR,2)
697       +     =(MW-LDCURVE(5))/(LDCURVE(6)-LDCURVE(5))*(0-.125)+.125
698             IF(MW.GT.LDCURVE(6))EQCURVE(NUMBERR,2)=0.
699  2005       CONTINUE
700             DO 5555 I5=1,100
701  5555       CONVLVD(I5)=0
702             DO 2205 IUNIT=1,NUNITS
703 C
704 C  LOOP 2205 IS A LOOP THAT CONVOLVES IN ALL OF THE CAPACITY STATES
705 C  FOR THE CURRENT EXPANSION PATTERN
706 C
707             IF(IUNIT.EQ.NUNITS)GOTO6220
708             INEXT=IUNIT+1
709 C
710 C  CHECK TO SEE IF THE NEXT CAPACITY STATE IN THE DISPATCH ORDER
711 C  IS FROM THE SAME GENERATOR.
712 C     IF YES, DON'T CONVOLVE IN THIS STATE.
713 C     IF NO, CHECK ANY PREVIOUSLY CONVOLVED STATES TO SEE IF THEY ARE
714 C    FROM THE SAME UNIT.
715 C
716             IF(XUNIT(IUNIT,1).EQ.XUNIT(INEXT,1).AND.XUNIT(IUNIT,2).EQ.
717       +     XUNIT(INEXT,2).AND.XUNIT(IUNIT,3).EQ.(XUNIT(INEXT,3)-1))
718       +     GOTO2205
719  6220       IF(XUNIT(IUNIT,3).EQ.1)GOTO6221
720                  DO 2235 IR=1,NUNITS
721                  IF(CONVLVD(IR).EQ.0)GOTO2235
722                  IF(XUNIT(IR,1).EQ.XUNIT(IUNIT,1).AND.XUNIT(IR,2).EQ.
723       +          XUNIT(IUNIT,2).AND.XUNIT(IR,3).EQ.(XUNIT(IUNIT,3)-1))
724       +          GOTO6235
725 C
726 C  CONVOLVE OUT THE SMALLER CAPACITY STATE OF THIS UNIT.
727  2235          CONTINUE
728             GOTO6221
729  6235           DO 3325 IPOINT=1,LASTPT
730                 NSTEPS=INT(IINDEX/UNIT(IR,1))
731                 OLDFRCT=EQCURVE(IPOINT,2)
732                      DO 3335 LSTEP=1,NSTEPS
```

```
733                     SLOPE1=(EQCURVE(IPOINT+1,2)-EQCURVE(IPOINT,2))/
734       +             IINDEX
735                     FRACT1=SLOPE1*LSTEP*UNIT(IR,1)+EQCURVE(IPOINT,2)
736                     FRACT2=(FRACT1-UNIT(IR,2)*OLDFRCT)/UNIT(IR,3)
737                     IF(LSTEP.EQ.NSTEPS)GOTO6331
738                     OLDFRCT=FRACT2
739  3335               CONTINUE
740  6331           SLOPE2=(FRACT2-OLDFRCT)/UNIT(IR,1)
741                 REMAIN=IINDEX-NSTEPS*UNIT(IR,1)
742                 EQCURVE(IPOINT+1,2)=SLOPE2*REMAIN+FRACT2
743  3325           CONTINUE
744             CONVLVD(IR)=0
745  6221       CONVLVD(IUNIT)=1
746 C
747 C  CONVOLVE IN THE CAPACITY STATE USING A DISCRETE CONVOLUTION
748 C
749                  DO 2215 IPOINT=1,LASTPT
750                  IPONT=LASTPT+1-IPOINT
751  2215            EQCURVE(IPONT,2)=UNIT(IUNIT,3)*EQCURVE(IPONT,2)+
752       +          UNIT(IUNIT,2)*XFRACT(UNIT(IUNIT,1),IPONT,IINDEX)
753  2205       CONTINUE
754 C
755 C  AFTER CONVOLVING IN THE LAST CAPACITY STATE, FIND THE LOLP
756 C
757             DO 2305 I=1,LASTPT
758             IF(ICAPSUM.LT.EQCURVE(I,1))GOTO6305
759  2305       CONTINUE
760  6305 XLOLP=(ICAPSUM-EQCURVE(I-1,1))/IINDEX*(EQCURVE(I,2)-EQCURVE(
761      +I-1,2))+EQCURVE(I-1,2)
762          IF(J.EQ.1)MAXLOLP=XLOLP
763          IF(J.EQ.2)AVELOLP=XLOLP
764    95 CONTINUE
765          RETURN
766   832 FORMAT(1X,I2,2X,F10.2)
767   930 FORMAT(A8,2(I2,2X),8X,F5.0,14X,F7.4)
768          END
769
770
771 C#########################################################################C
772 C#########################################################################C
773 C                                                                         C
774 C                            FUNCTION XFRACT                              C
775 C                                                                         C
776 C#########################################################################C
777 C#########################################################################C
778
779
780       FUNCTION XFRACT(UNCAP,IPOINT,IINDEX)
781 C
782 C  FUNCTION XFRACT CALCULATES THE CORRESPONDING PROBABILITY FOR ANY
783 C  (MW-UNCAP) VALUE FROM THE CURRENT LOAD PROBILITY DISTRIBUTION
784 C  CURVE
785 C
786       DIMENSION I(4),ENTRY(30,30,4),ALSOMAX(20,4),INDEX(100),
787      +NAME(100),NUMBER(100),UNIT(100,3),EQCURVE(200,2),IBESTOP(5,30),
788      +HTRATE(7),CAPCTY(7),IFUEL(100),MLTSTAT(100),ECO(30),BENEFIT(30,
789      +5,2)
790       REAL LDORDER(100
791      +,5),LDCURVE(7),MAXRET(5,30),ECTRY(30,30,4),ICAPSUM,IINDEX,LOADEQ
792       INTEGER YEAR,CAPTLMX,OPTION(4,6),XOPTION(30),STATE(30),BCOUNT,
793      +ECOUNT,COUNT,CONVLVD(100),XUNIT(100,3),NFLAG(100),CSTATE(100),
```

```
      *HIGUES
       COMMON I,ENTRY,CAPCTY,UNIT,EQCURVE,IBESTOP,ALSOMAX,BENEFIT,
      +LDORDER,MAXRET,XOPTION,CSTATE,IFUEL,MLTSTAT,ECO,ECTRY,
      +NFLAG,XUNIT,OPTION
       EQUIVALENCE (XUNIT(1,1),NAME),(XUNIT(1,2),NUMBER),(XUNIT(1,3),
      +INDEX),(LDCURVE,HTRATE,CAPCTY),(CONVLVD,NFLAG),(STATE,XOPTION)
C
C  EQCURVE IS THE ARRAY THAT HOLDS THE DISCRETE POINTS THAT APPROXIMATE
C  THE LOAD PROBABILITY DISTRIBUTION CURVE.  THE CURVE IS APPROXIMATED
C  PIECE-WISE LINEAR BETWEEN THESE POINTS.  IINDEX IS THE MW DIFFERENCE
C  BETWEEN THESE POINTS.
C
       LOADEQ=EQCURVE(IPOINT,1)-UNCAP
C
C  LOADEQ=IPOINT'S MW VALUE - THE UNIT CAPACITY
C  EQCURVE(IPOINT,1) HOLDS THE MW VALUE OF PT. IPOINT
C  EQCURVE(IPOINT,2) HOLDS THE PROBABILITY OF PT. IPOINT
C
       IF(LOADEQ.LE.EQCURVE(1,1))GOTO6001
C
C  IF LOADEQ IS LESS THAN THE BASE LOAD, THE PROB.=1
C
       LOWGUES=INT((LOADEQ-EQCURVE(1,1))/IINDEX)
 6006  HIGUES=LOWGUES+1
       IF(LOADEQ-EQCURVE(HIGUES,1))6002,6002,6004
C
C  CHECK TO SEE IF THE LOAD IS BETWEEN SOME LOW GUESS AND HIGH GUESS
C
 6002  IF(EQCURVE(LOWGUES,2).EQ.0..AND.EQCURVE(HIGUES,2).EQ.0.)GOTO6007
C
C  IF LOADEQ IS GREATER THAN THE EQUIVALENT PEAK LOAD, THE PROB.=0.0
C
       IF(EQCURVE(LOWGUES,2).NE.0..AND.EQCURVE(HIGUES,2).EQ.0.)GOTO6005
C
C  IF LOADEQ IS BETWEEN THE EQUIV. PEAK LOAD AND THE NEXT TO LAST PT.
C  GO TO STATEMENT 6005
C    ELSE, CALCULATE THE PROBABILITY BY INTERPOLATING
C
       XFRACT=(LOADEQ-EQCURVE(LOWGUES,1))/IINDEX*(EQCURVE(HIGUES,2)-
      +EQCURVE(LOWGUES,2))+EQCURVE(LOWGUES,2)
       RETURN
 6004  LOWGUES=LOWGUES+1
       GO TO 6006
 6001  XFRACT=1.0
       RETURN
 6005  EQMAXLD=EQCURVE(LOWGUES,1)-((EQCURVE(LOWGUES,1)-EQCURVE(LOWGUES-1
C
C  USING THE PREVIOUS TWO PTS. OF THE CURVE (LOWGUES AND LOWGUES-1)
C  CALCULATE WHERE THE LINE CONNECTING THESE TWO POINTS HITS THE LOAD
C  AXIS.  IF LOADEQ IS GREATER THAN THIS LOAD VALUE, THE PROB=0.0 .
C     ELSE, INTERPOLATE USING LOWGUES AND THE POINT EQMAXLD TO FIND
C THE PROBABILITY
C
      +,1))/(EQCURVE(LOWGUES,2)-EQCURVE(LOWGUES-1,2))*EQCURVE(LOWGUES
      +,2))
       IF(LOADEQ.GE.EQMAXLD)GOTO6007
       XFRACT=(LOADEQ-EQCURVE(LOWGUES,1))/(EQMAXLD-EQCURVE(LOWGUES,1))
      +*(0.-EQCURVE(LOWGUES,2))+EQCURVE(LOWGUES,2)
       RETURN
 6007  XFRACT=0.
       RETURN
```

```
      END


C#######################################################################C
C#######################################################################C
C                                                                       C
C                      SUBROUTINE RELBLTY                               C
C                                                                       C
C#######################################################################C
C#######################################################################C


      SUBROUTINE RELBLTY(MAXLOLP,AVELOLP,RELBNFT,CLOLP,WK1,WK2)
C
C  THIS SUBROUTINE CALCULATES RELIABILITY BENEFIT AS A FUNCTION OF
C  THE AVERAGE AND MAXIMUM WEEKS' LOLP
C
      REAL MAXLOLP
      RELBNFT=WK1*EXP(-MAXLOLP*CLOLP)+WK2*EXP(-AVELOLP*CLOLP)
C
C  THE RELIABILITY BENEFIT IS CALCULATED FROM THE UTILITY FUNCTION
C  USING THE USER SPECIFIED WEIGHTING CONSTANTS AND EXP. CONSTANT
C
      RETURN
      END


C#######################################################################C
C#######################################################################C
C                                                                       C
C                      SUBROUTINE MAXSTAT                               C
C                                                                       C
C#######################################################################C
C#######################################################################C


      SUBROUTINE MAXSTAT(KOUNT,MLTFLAG,COUNT)
C
C  SUBROUTINE MAXSTAT CALULATES THE MULTI-ATTRIBUTE UTILITY OF ALL
C  POSSIBLE STATES IN EVERY YEAR.  THE BEGINNING STATES THAT GIVE THE
C  MAXIMUM RETURN FOR EACH EXPANSION OPTION ARE SAVED IN THE IBESTOP
C  ARRAY, AND THE MAXIMUM RETURN IS SAVED IN THE MAXRET ARRAY
C
      DIMENSION I(4),ENTRY(30,30,4),ALSOMAX(20,4),INDEX(100),
     +NAME(100),NUMBER(100),UNIT(100,3),EQCURVE(200,2),IBESTOP(5,30),
     +HTRATE(7),CAPCTY(7),IFUEL(100),MLTSTAT(100),BENEFIT(30,5,2),
     +ECO(30)
      REAL LDORDER(100
     +,5),LDCURVE(7),MAXRET(5,30),ECTRY(30,30,4),ICAPSUM,N
      INTEGER YEAR,CAPTLMX,OPTION(4,6),XOPTION(30),STATE(30),BCOUNT,
     +ECOUNT,COUNT,CONVLVD(100),XUNIT(100,3),NFLAG(100),CSTATE(100)
      COMMON I,ENTRY,CAPCTY,UNIT,EQCURVE,IBESTOP,ALSOMAX,BENEFIT,
     +LDORDER,MAXRET,XOPTION,CSTATE,IFUEL,MLTSTAT,ECO,ECTRY,
     +NFLAG,XUNIT,OPTION
      EQUIVALENCE (XUNIT(1,1),NAME),(XUNIT(1,2),NUMBER),(XUNIT(1,3),
     +INDEX),(LDCURVE,HTRATE,CAPCTY),(CONVLVD,NFLAG),(STATE,XOPTION)
      Y="Y"
      N="N"
C
C  ASK THE USER IF SHE WANTS TO ENTER WEIGHTING COEFFICIENTS
C
```

```
916     62 WRITE(6,900)
917        WRITE(2,900)
918        READ(7,800)X
919        COUNT=0
920        IF(X.EQ.Y)GOTO61
921        IF(X.EQ.N)RETURN
922 C
923 C   THE ANSWER MUST BE YES OR NO
924 C
925        WRITE(6,901)
926        WRITE(2,901)
927        GOTO62
928     61 WRITE(6,902)
929        WRITE(2,902)
930 C.
931 C   HAVE THE USER ENTER THE COEFFICIENTS, AND READ THEM
932 C
933        READ(7,801)W1,W2,W3
934        WRITE(2,924)W1,W2,W3
935 C
936             DO 205 YEAR=1,5
937             IF(YEAR.NE.1)GOTO6207
938                DO 105 IOPTION=1,KOUNT
939                MAXRET(YEAR,IOPTION)=W1*BENEFIT(IOPTION,YEAR,1)+W2*BENEFIT(
940       +        IOPTION,YEAR,2)+W3*ECO(IOPTION)
941    105         IBESTOP(YEAR,IOPTION)=1
942                GOTO6101
943 C
944 C   CALCULATE THE RETURNS FOR EVERY POSSIBLE EXPANSION STATE
945 C    FOR EVERY POSSIBLE BEGINNING STATE FOR THIS STAGE. (YEAR)
946 C   USING THESE WEIGHTING COEFFICIENTS
947 C
948   6207          DO 215 IOPTION=1,KOUNT
949                      DO 225 NSTATE=1,KOUNT
950                      ENTRY(NSTATE,IOPTION,YEAR-1)=W1*BENEFIT(IOPTION,YEAR
951       +              ,1)+W2*BENEFIT(IOPTION,YEAR,2)+W3*ECTRY(NSTATE,
952       +              IOPTION,YEAR-1)+MAXRET(YEAR-1,NSTATE)
953    225               IF(ECTRY(NSTATE,IOPTION,YEAR-1).EQ.0.0)ENTRY(
954       +              NSTATE,IOPTION,YEAR-1)=0.0
955 C
956 C   IF THE ECONOMIC RETURN IS 0.0, THEN THE EXPANSION STATE DOES NOT
957 C   MEET THE CAPITAL CONSTRAINT SO MAKE THE MULTI-ATTR. UTILTY 0.0
958 C
959                MAXRET(YEAR,IOPTION)=ENTRY(1,IOPTION,YEAR-1)
960                IBESTOP(YEAR,IOPTION)=1
961                     DO 235 ISTATE=2,KOUNT
962                     IF(ENTRY(ISTATE,IOPTION,YEAR-1)-MAXRET(YEAR,IOPTION)
963       +             )235,6231,6232
964 C
965 C   CHECK THE RETURN FOR THE STATE UNDER CONSIDERATION AGAINST THE
966 C   CURRENT MAXIMUM RETURN.  IF IT IS SMALLER, DO NOT SAVE THIS AS THE
967 C   NEW MAXIMUM.  IF EQUAL TO THE MAXIMUM RETURN SAVE IT AS A DUPLICATE
968 C   MAXIMUM.  IF IT IS GREATER THAN THE CURRENT MAXIMUM, SAVE IT AS THE
969 C   NEW MAXIMUM RETURN AND GET RID OF ANY DUPLICATES OF THE PREVIOUS
970 C   MAXIMUM RETURN.  ALSO SAVE THE BEGINNING STATE OF THE STAGE THAT
971 C   HAS THIS MAXIMUM RETURN.
972 C
973   6231          COUNT=COUNT+1
974 C
975 C   COUNT KEEPS TRACK OF THE NUMBER OF DUPLICATE MAXIMUM STATES
976 C
```

```
977                ALSOMAX(COUNT,1)=YEAR
978                ALSOMAX(COUNT,2)=XOPTION(IOPTION)
979                ALSOMAX(COUNT,3)=MAXRET(YEAR,IOPTION)
980                ALSOMAX(COUNT,4)=XOPTION(ISTATE)
981 C
982 C  ALSOMAX IS THE ARRAY THAT HOLDS THE INFORMATION ABOUT DUPLICATE
983 C  MAXIMUM STATES
984 C
985                GOTO235
986  6232          IF(COUNT.EQ.0)GOTO6234
987                KKOUNT=0
988 C
989 C  THE NEXT SEVEN LINES GET RID OF ANY DUPLICATES OF THE OLD MAXIMUM
990 C  STATE, IF THERE WERE ANY.
991 C
992                     DO 245 IALSO=1,COUNT
993                     IF(ALSOMAX(IALSO,2).NE.XOPTION(IOPTION))GOTO245
994                     IF(ALSOMAX(IALSO,3)-ENTRY(ISTATE,IOPTION,YEAR-1)
995      +              )6241,245,245
996  6241               KKOUNT=KKOUNT+1
997   245               CONTINUE
998                COUNT=COUNT-KKOUNT
999 C
1000 C  SAVE THE MAXIMUM RETURN FOR THIS EXPANSION STATE, AS WELL AS THE
1001 C  BEGINNING STATE THAT LED TO THIS EXPANSION STATE.
1002 C
1003  6234          MAXRET(YEAR,IOPTION)=ENTRY(ISTATE,IOPTION,YEAR-1)
1004                IBESTOP(YEAR,IOPTION)=ISTATE
1005   235          CONTINUE
1006   215     CONTINUE
1007 C
1008 C  WRITE OUT THE RETURNS FOR ALL STATES IN ALL YEARS AS WELL AS THE
1009 C  WEIGHTING COEFFICIENTS THAT WERE USED IN THE MULTI-ATTRIBUTE UTILITY
1010 C  FUNCTION
1011 C
1012  6101     WRITE(2,919)YEAR
1013           WRITE(2,940)
1014           WRITE(2,922)(XOPTION(N1),N1=1,KOUNT)
1015           WRITE(2,941)
1016           WRITE(2,942)
1017           IF(YEAR.EQ.1)WRITE(2,920)STATE(YEAR),(MAXRET(YEAR,NDEC),
1018      +    NDEC=1,KOUNT)
1019           IF(YEAR.EQ.1)GOTO6201
1020                DO 115 NSTATE=1,KOUNT
1021   115           WRITE(2,920)STATE(NSTATE),(ENTRY(NSTATE,NDEC,YEAR
1022      +          -1),NDEC=1,KOUNT)
1023  6201 WRITE(2,927)
1024       WRITE(2,928)(XOPTION(LL),LL=1,KOUNT)
1025           WRITE(2,921)(MAXRET(YEAR,IOPTION),IOPTION=1,KOUNT)
1026           WRITE(2,923)(XOPTION(IBESTOP(YEAR,IOPTION)),IOPTION=1,KOUNT)
1027   205     CONTINUE
1028       IF(COUNT.GT.0)MLTFLAG=1
1029 C
1030 C  MLTFLAG IS A FLAG THAT IS SET =1 IF THERE ARE SOME DUPLICATE MAX-
1031 C  IMUM STATES.
1032 C
1033       IF(MLTFLAG.EQ.1)WRITE(2,931)MLTFLAG
1034       IF(MLTFLAG.EQ.1)WRITE(2,935)
1035       IF(MLTFLAG.EQ.1)WRITE(2,933)((ALSOMAX(IT,IS),IS=1,4),IT=1,KCOUNT)
1036       IBEST=1
1037       VALUE=MAXRET(5,1)
```

```
1038            DO 545 IS=2,KOUNT
1039            IF(MAXRET(5,IS)-VALUE)545,545,6541
1040  6541      IBEST=IS
1041            VALUE=MAXRET(5,IS)
1042   545      CONTINUE
1043        WRITE(2,925)XOPTION(IBEST),VALUE
1044        GOTO62
1045   800 FORMAT(A1)
1046   801 FORMAT(3(F6.4,1X))
1047   900 FORMAT("-","DO YOU WANT TO ENTER WEIGHTING COEFFICIENTS? (TYPE Y"
1048      +" FOR YES, N FOR NO",/)
1049   901 FORMAT(1X,"ERROR:  YOU MUST TYPE EITHER Y OR N TO CONTINUE!")
1050   902 FORMAT(1X,"TYPE IN THE ENVIRONMENTAL, RELIABILITY AND ECONOMIC "
1051      +"WEIGHTING FACTORS IN THE FORMAT 3(F6.4,1X)  EXAMPLE: ",/,
1052      +"0.2500 0.5000 0.2500",/)
1053   919 FORMAT("1",9X,"**********************************",
1054      +"********** THE RETURN VALUES FOR YEAR ",I2," ***************",
1055      +"****************************")
1056   920 FORMAT(1X,3X,I4,2X,30F4.2,/)
1057   921 FORMAT("-","MAX RTRN ",30F4.2)
1058   922 FORMAT(1X,"STATES-->",30I4,/,/)
1059   923 FORMAT("-","BEG STATE",30I4)
1060   924 FORMAT(" ",9X,"ENVIRONMENTAL WT.=",F6.4,"  RELIABILITY WT.=",F6.4,
1061      +"  ECONOMIC WT.=",F6.4)
1062   925 FORMAT("-","********* THE BEST EXPANSION OPTION IN YEAR 5 IS ",
1063      +"STATE:",I5," THE OVERALL UTILITY FOR THE OPTIMAL PATH IS:",
1064      +F10.7,/,/,/,/)
1065   927 FORMAT(" ",130(1H-))
1066   928 FORMAT("-","XPN STATE",30I4)
1067   931 FORMAT(1X,"STATE   MAXIMUM RETURN   BEST DECISION   MULTFLAG=",I2)
1068   933 FORMAT(1X,3X,I2,7X,I2,8X,I2,9X,I2)
1069   935 FORMAT(1X,"STAGE STATE   RETURN   DECISION")
1070   940 FORMAT(1X,"EXPANSION")
1071   941 FORMAT(1X,"BEGINNING")
1072   942 FORMAT(1X,"  STATES ",/)
1073        END
```

Two test systems were used as input for an expansion planning simulation. System A consisted of all generating units and input information contained in Appendix B. System B was identical to system A except plant K, unit 4 and plant E, unit 6 were not included. Also the load growth rate was set to 10% and the capital escalation rate was set equal to the construction cost escalation rate of 8%. The results of the test system are summarized in Table C.1.

TABLE C.1

TEST RESULTS

Weighting Coefficients			Final State of "Optimal" Plan	
			System A	System B
Env.	Rel.	Eco.		
0.2000	0.6000	0.2000	0000	0022
0.3333	0.3333	0.3333	0000	0000
0.0000	0.8000	0.2000	1002	1002
0.2500	0.5000	0.2500	0000	0002
0.0000	0.2000	0.8000	0002	0002

The execution time for both programs combined was in the range of 90-100 seconds when thirty expansion options were considered. When only ten expansion options were considered, the execution time was in the range of 40-45 seconds.

REFERENCES

[1] K. M. Dale, "Dynamic Programming Approach to the Selection and Timing of Generating Plant Additions," Proc. IEE, vol. 113, no. 5, pp. 803-811, May 1966.

[2] E. N. Oatman and L. J. Hamant, "A Dynamic Approach to Generation Expansion Planning," IEEE T-PAS, vol. PAS-92, pp. 1888-1897, November/December 1973.

[3] W. F. Esser, P. Ghose, and K. Chen, "Decision Analysis for Electric Power Systems Engineering and Management," IEEE T-PAS, vol PAS-96, no. 2, pp. 447-456, March/April 1977.

[4] J. J. Bolinger Jr., P. Ghose, J. H. Sosinski, and W. F. Esser, "Decision Analysis Utilizing Multi-Attribute Utility Theory in Engineering Evaluations,: IEEE T-PAS, vol. PAS-97, no. 4, pp. 1245-1253, July/August 1978.

[5] R. T. Jenkins and D. S. Joy, WEIN Automatic System Planning Package (WASP) - An Electric Utility Optimal Generation Expansion Planning Computer Code, Oak Ridge National Laboratory, ORNL-4945, July 1974.

[6] R. L. Keeney and H. Raiffa, Decisions With Multiple Objectives. New York: John Wiley & Sons, 1976.

[7] R. L. Sullivan, Power System Planning. New York: McGraw-Hill, 1977.

[8] R. Bellman, Dynamic Programming. Princeton: Princeton University Press, 1957.

[9] R. T. Larson, State Increment Dynamic Programming. New York: American Elsevier Publishing Company, 1968.